AN A PRIORI MULTIOBJECTIVE OPTIMIZATION MODEL OF A SEARCH AND RESCUE NETWORK

THESIS

Joseph C. Imsand, Jr., Captain, USAF

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GOR/ENS/92M-17

AN *A PRIORI* MULTIOBJECTIVE OPTIMIZATION MODEL OF A SEARCH AND RESCUE NETWORK

THESIS

Joseph C. Imsand, Jr., Captain, USAF

AFIT/GOR/ENS/92M-17

AN A PRIORI MULTIOBJECTIVE OPTIMIZATION MODEL OF A SEARCH AND RESCUE NETWORK

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Operations Research

Joseph C. Imsand, Jr., B.S.

Captain, USAF

March 1992

Approved for public release; distribution unlimited

THESIS APPROVAL

STUDENT: Captain Joseph C. Imsand, Jr.   CLASS: GOR-92M

THESIS TITLE: An *A Priori* Multiobjective Optimization Model of a Search and Rescue Network

DEFENSE DATE: 3 March 1992

GRADE:

COMMITTEE:	NAME/DEPARTMENT	SIGNATURE
Advisor	Dr Yupo Chan (AFIT/ENS)	Yupo Chan
Reader	Dr James Chrissis (AFIT/ENS)	J W C

Acknowledgements

I would like to thank several people for their help and guidance during this research process. First, I want to thank my advisor, Dr. Yupo Chan, for his patience and direction in keeping me on the right track. Next, I want to thank Dr. James Chrissis, Dr. Alfred Marsh and Mr. Chris Dearing for all of their helpful insight and comments over the past few months. I would also like to thank Capt Krista Johnson and Capt Jean Steppe for sharing their lessons learned from previous research in this area. Last, but certainly not least, I want to thank my wife, Deidra, my children, Joey and Kayla, and my parents for all their support in keeping me sane and focused during the last several months.

Joseph C. Imsand, Jr.

Table of Contents

List of Figures

List of Tables

Abstract

The purpose of this study was to determine the most robust frequency assignment for a search and rescue network. The focus was to assign weights to the transmitter areas of the network to determine which weight sequence produced the most robust frequency assignment. The Department of Defense furnished weight sequences for twelve two-hour time blocks. These weight sequences were compared to a weight sequence with all weights of equal value. Network and linear programming were used to solve this problem and generate frequency assignments for all weight sequences. Classical sensitivity analysis and tolerance analysis were used to analyze the frequency assignments generated by the different weight sequences. The weight sequence with all weights having equal value produced the most robust frequency assignments for all time blocks.

An *A Pricri* Multiobjective

Optimization Model of a

Search and Rescue Network

I. Introduction

1.1 Background

The United States currently operates a network of search and rescue (SAR) stations around the world over extensive ocean areas. These stations receive and process distress signals from airplanes or ships that experience emergencies. SAR missions occur only when three or more stations receive and process the same distress signal because it takes at least three stations to geolocate the source of the signal (7:1).

Each station has two types of systems that perform the geolocation process. The receiving subsystem's (RS) primary mission is detecting the distress signal and initiating the geolocation process. On the other hand, the high frequency direction finding subsystem (HFDF) estimates the location of the signal transmitter in association with the RS (7:1).

Signal acquisition by the RS is the sole element in starting the successful geolocation of a distress signal. When a station acquires a signal via the RS, the station operator alerts Central Control (CC) of the acquisition. CC

then prompts the other stations requesting lines of bearing (LOB) for the signal of interest. The other operators check their RS and HFDF systems for the signal of interest and transmit a LOB back to CC if available. CC uses the LOBs to compute a best point estimate and a confidence region for geolocation of the signal transmitter. If a HFDF system receives a signal but a RS system does not, no attempt at geolocation is made because HFDF systems cannot notify CC of the signal (7:1).

Every station in the network contains one RS system, but the number of HFDF receivers at each station varies. The minimum number of HFDFs is 0 and the maximum number is 10. The RS samples the entire frequency spectrum of interest but is less sensitive than the HFDF. The RS cannot acquire signals with small signal-to-noise ratio like the HFDF can. Each HFDF receiver covers a 1 MHz band within the frequency spectrum. These HFDF receivers are optimally allocated to frequencies at the stations so that the probability of successfully geolocating a vehicle in distress is maximized (7:1).

Prior research efforts in this area have produced optimal methods to locate the stations and the frequency assignments. Steppe used a two-stage, network-flow multiobjective linear integer programming (MOLIP) model to determine the optimal locations of the stations for the SAR problem (15). Another

effort in this area by Johnson found the optimal frequency assignments using a MOLIP network-flow model (10).

1.2 Research Objective   The purpose of this research is to use a priori optimization to show that the optimal assignment of HFDF receivers in a generalized search and rescue (GSAR) network is independent of the weighting of the transmitter areas. This is accomplished by investigating the effect of changing the weight value of a particular transmitter area on the probability of geolocation for that area.

1.3 Overview

The next chapter is a literature review covering a priori optimization, network programming and sensitivity analysis. Chapter three describes a network model and a linear program model formulation for this problem. The next chapter contains a sample problem formulation and solution. Chapter four describes the solution methodology for both the network and linear program models and the sensitivity analysis to be performed. The last chapter lists the results from the solution methodologies, draws some conclusions from the results and makes some recommendations for future research.

II. Literature Review

This chapter presents an overview of the literature that will contribute to the formulation of this problem. A general overview of networks is provided in the first section. The second section will describe a particular type of network that will be used to formulate this problem. The third section will cover *a priori* optimization and how it can be applied to four different combinatorial optimization problems. The fourth and final section will cover sensitivity analysis and its value for linear programming and network programming problems.

2.1 Networks.

A network consists of a set of nodes connected by a set of arcs where the arcs represent flow between the nodes. Examples of networks are highway intersections, telephone exchanges, and airline terminals (12:6). The corresponding arcs are roads, telephone lines, and airline routes (12:6). The arc flow is usually constrained by the capacity each arc can handle. The nodes are constrained by the *conservation of flow*. Conservation of flow means that the total flow into a node must equal the total flow out of that node. Minimizing

the cost and maximizing the flow are two objectives sometimes associated with network programming problems.

2.2 Maximum-Flow Network.

The frequency assignment of HFDF receivers in a search and rescue network can be formulated and solved as a maximum-flow network. The next section discusses the concepts of a maximum-flow network.

The maximum-flow problem is equivalent to a directed, connected network with one supply node and one demand node. The other nodes are considered transshipment nodes which simply means they preserve conservation of flow. The objective is to maximize the total flow through the network from the supply node to the demand node subject to the arc capacities given for each arc and the cost associated with each arc. The assignment of HFDF receivers is represented by a set of nodes corresponding to the receivers and a set of nodes corresponding to the frequencies to be covered. The arcs between the two sets of nodes have a capacity of one, meaning one receiver can cover one frequency. Also, the arcs have a cost which represents the probability of detecting a signal of interest (8:359-366).

2.3 A Priori Optimization.

Bertsimas, Jaillet, and Odoni describe _a priori_ optimization as "a strategy competitive to the strategy of reoptimization, under which the combinatorial optimization problem is solved optimally for every instance" (6:1019). _A priori_ optimization consists of two parts. The first part is a measure of effectiveness and the second part is a method to update the _a priori_ solution for each problem occurrence (6:1020). Bertsimas, Jaillet, and Odoni made a logical choice of the expected cost as the measure of effectiveness. They also provide the following three properties necessary for the updating procedure:

> First, for every choice of an updating procedure proposed, the updating of the solution to a particular instance can be done very easily. Next, these updating methods are well suited for applications. And finally, the _a priori_ optimization strategies coupled with the particular choices of the updating procedure are asymptotically very close in terms of performance to the reoptimization strategies under reasonable probabilistic assumptions. (6:1020)

Berman and Simchi-Levi introduce two motivations for this type of problem. First, a company may not be able to reoptimize every day because some demands may not be known until just before the work day begins (3:148). The second reason the company may not want to reoptimize is that the cost may be too high (3:148).

2.3.1 Probabilistic Traveling Salesman Problem. The first combinatorial optimization problem where a priori optimization is used is the probabilistic traveling salesman problem (PTSP). The traveling salesman problem (TSP) is one of the most extensively covered problems in optimization (9:929). It deals with finding the minimum distance a salesman travels while visiting each of a set of n cities. In the TSP, the salesman visits every city in his area of responsibility on each trip. Therefore, the TSP is a deterministic problem. On the other hand, in the PTSP, the salesman visits only k out of the n cities on any given trip (9:930). The cities have probability p_i of being visited on each trip. The probabilities are independent and are not necessarily equal.

Jaillet defines the PTSP as "finding an a priori tour of minimum length in the expected value sense" (9:929). In order to solve this problem, an a priori tour through all n points is determined. For any new tour of k points, the solution for this new tour will visit the k points in the same order as the a priori tour of n points (6:1020). The updating procedure is to visit the k points for every possible tour combination in the same sequence as the a priori tour (6:1020). As Merrill, Chan, and Schuppe deduced from Hardgrave and Nemhauser, "the

order of visitation around the convex hull circumscribing a network does not change in reaching the optimal solution" (11:6).

2.3.2 Probabilistic Minimum Spanning Tree Problem. *A priori* optimization is also applicable to a problem known as the probabilistic minimum spanning tree problem (PMSTP). The classical minimum spanning tree problem finds a tree that connects all *n* network nodes with no cycles and has the shortest overall length possible. In contrast, in the PMSTP only a subset *S* of the *n* nodes is present at any given time with probability *p(S)* (5:245).

Bertsimas describes the PMSTP as "finding an *a priori* spanning tree of minimum expected length over all possible problem instances" (5:245). In order to accomplish this, an *a priori* spanning tree must be found. This tree is used as follows:

> On any given instance of the problem, the *a priori* tree is retraced deleting only the nodes that are not present, provided the deletion of those nodes does not disconnect the tree. In this way, there would be nodes that will not be present but are still included in the tree. Thus, the updating method is to include all nodes in the instance *S* and also those other nodes in the network that are necessary to prevent the resulting tree from becoming disconnected. (6:1021)

2.3.3 Probabilistic Vehicle Routing Problem. The third area where *a priori* optimization is used is the probabilistic vehicle routing problem (PVRP). Like the previous two problems, the stochastic problem is an extension of the more familiar deterministic vehicle routing problem (VRP). The VRP contains *n* cities or nodes, all of which must be visited. The solution is a set of routes with minimum length covering all *n* nodes. On the other hand, the PVRP contains nodes with probabilistic, rather than deterministic, demands (6:1021).

Bertsimas, Jaillet, and Odoni describe the PVRP as "determining a fixed set of routes of minimal expected total length, which corresponds to the expected total length of the fixed set of routes plus the expected value of extra travel distance that might be required" (6:1021). The extra distance occurs when the demand of a route exceeds the capacity of the vehicle, a condition which forces the vehicle to return to the warehouse to reload before continuing the route (6:1021). Two updating procedures are proposed by Bertsimas, Jaillet, and Odoni. In the first procedure, the designated vehicle visits every point in the same order as the *a priori* tour. However, the vehicle only provides service to the points that have a requirement during a particular occurrence (6:1021). The second procedure is very similar to the first. The only difference is a point with no demand during a particular

occurrence is skipped entirely on the *a priori* tour and the next point with a demand is serviced (6:1021).

2.3.4 Probabilistic Traveling Salesman Facility Location Problem. The final problem discussed in relation to *a priori* optimization is the probabilistic traveling salesman facility location problem (PTSFLP). This problem is a traveling salesman facility location problem with demands for service represented by a probability p_i. The problem consists of a network with a set of n nodes that need service and a single service facility. A service unit leaves the service facility each day at a given time and proceeds to the nodes that have a demand for that day (14:479). It is assumed that only one call can arrive from a given node on a given day. Hence, the maximum number of calls on any day is n (14:479). Also, the probability of a demand from a given node is assumed to be independent of the probability of a demand from a different node (14:479).

Bertsimas describes the PTSFLP as "the problem of finding simultaneously an optimal location for the service facility and an optimal *a priori* tour" (4:184). The optimal *a priori* tour for this problem is calculated using the probability for each node. The updating procedure is the same as for the PTSP

described above. A node is skipped on the a priori tour when no demand exists for the node for a particular occurrence (4:185).

2.4 Sensitivity Analysis.

The purpose of sensitivity analysis is to determine how much the parameters of the model can be changed without changing the optimal solution. This is important because in many models the coefficients of the parameters are estimates whose true values are not known. Therefore, if one of the estimates is not accurate and the optimal solution is very sensitive to that parameter the given optimal solution may not be the real optimal solution. The rest of this section discusses how sensitivity analysis evaluates changes in the coefficients of a variable and in the right hand side values of the constraints. It also shows how a new constraint and a new variable may affect an optimal solution.

2.4.1 Change in Variable Coefficients.

There are two cases to be examined when a variable has one or more coefficients changed. The first case is when the coefficients of a nonbasic variable of the optimal solution are changed. The coefficients that may be changed are the objective function coefficients or the constraint coefficients. An easy method of determining if the solution is still optimal is to check that the dual problem complementary solution still

satisfies the single dual constraint that was changed. If it is satisfied, the solution is still optimal. On the other hand, if the dual constraint is violated, the column of the optimal tableau that has been changed must be recalculated using the new numbers, the reduced cost, $\underline{C_b B^{-1} a_i - c_i}$, value that is positive enters the basis and the simplex method is applied so that a new optimal solution is determined (8:175-176).

The second case is when the coefficients of a basic variable are changed. In this case, the new column of the optimal tableau is calculated and the tableau is put in proper form via Gaussian elimination. If the new tableau is either not optimal or infeasible, the simplex or dual-simplex method is applied to produce a new optimal solution. However, if the tableau is still optimal no further calculations are required (8:175-177).

<u>2.4.2 Change in the Right-Hand-Side Value.</u> When the right-hand-side of one or more constraints is changed, the right-hand-side value column of the optimal tableau is recalculated. Because this is the only change in the tableau, Gaussian elimination is not required and optimality does not need to be tested. But, the feasibility of the solution must be examined. If it is optimal no further calculations are required. However, if the solution is infeasible, the dual-

simplex method must be applied to produce a new optimal feasible solution (8:172-174).

2.4.3 Add a New Variable. When a new variable is added to a model after the optimal solution has been found, a new column is added to the tableau. The easiest way to handle this situation is to treat it like a nonbasic variable whose original cost coefficients are zero. Therefore, check if the complementary basic solution satisfies the new dual constraint. If it does, the solution is still optimal. If it does not, use the same procedure as above for the nonbasic variable with different coefficients to determine the new optimal solution (8:176-177).

2.4.4 Add a New Constraint. The quickest way to determine if the optimal solution is still valid when a new constraint is added is to check if the current solution satisfies the new constraint. If it does, then no calculations are required. If it does not, a new row representing the new constraint is added to the optimal tableau. The tableau is put into proper form via Gaussian elimination and the dual simplex method is applied to the new tableau and a new optimal solution is obtained (8:180).

2.4.5 The Tolerance Approach. Another method employed in sensitivity analysis is called the tolerance approach. This approach differs from the methods of sensitivity analysis discussed previously in that it deals with simultaneous and independent perturbations of the objective function coefficients and the right-hand-side values. It produces a maximum tolerance percentage for both the objective function coefficients and the right-hand-side values. This tolerance percentage is the maximum percentage any value can vary from its original estimated value while retaining the same optimal solution. The rest of this section covers how the tolerance approach can be applied to linear programming and network programming problems.

2.4.5.1 Linear Programming. For linear programming, the tolerance approach uses information from the original tableau and the optimal tableau in order to determine T^* and R^*. T^* is the maximum tolerance percentage for the objective function coefficients and R^* is the maximum tolerance percentage for the right hand side values. The equation for calculating T^* is

$$T^* = \underset{k \in K}{Min} \left[\frac{C_b B^{-1} A_{.k} - C_k}{|C_k| + \sum_{i=1}^{m} |C_{hi} B_{i.}^{-1} A_{.k}|} \right].$$

The set {K} contains the subscripts of all the nonbasic variables. The numerator is the reduced cost of X_k, c_k is the cost coefficient of the <u>k</u>th variable, and $B^{-1}_{i.}A_{.k}$ is the <u>i</u>th term in the <u>k</u>th column of the optimal tableau. Also, c_h is the coefficient of the <u>i</u>th variable of the cost vector. If the denominator for any term is zero, then the value for that term is infinity. Additionally, if $T^{*}= 0$ then there may be alternate optimal solutions (16:566-567).

The maximum allowable tolerance for the right hand side values is determined by the following equation:

$$R^{*} = \min_{k=1,\ldots,m} \left[\frac{B_{k.}^{-1} b}{\sum_{j=1}^{m} |B_{kj}^{-1} b_j|} \right].$$

The numerator is the <u>k</u>th element of the right-hand-side column of the optimal tableau and B^{-1}_{kj} is the <u>k</u>th element of the <u>j</u>th column of the optimal tableau and b_j is the <u>j</u>th element of the original right-hand-side vector. Again, if the denominator of any term is zero then the value for that term is infinity and if $R^{*}= 0$ then an alternate optimal solution is possible (16:566-567).

<u>2.4.5.2 Network Programming.</u> In network programming, the tolerance approach is adapted from the linear programming approach so that computations can be performed

directly on the network. The equation for T^* is

$$T^* = \underset{k \in J}{Min} \left[\frac{|c_k|}{\sum_{i \in S_k} |c_i|} \right].$$

"If J is empty, then T^* is infinite. If J is not empty then T^* is finite iff for some k an element of J there exists an edge e_k with a nonzero coefficient c_k in the fundamental cycle corresponding to the chord e_k. The set S_k is the index set of the edges in the fundamental cycle corresponding to the chord e_k" (13:164).

The corresponding equation for the right hand side values is

$$R^* = \underset{k=1,\ldots,m}{Min} \left[\underset{\substack{|b_h| \geq 0 \\ h=1,\ldots,m}}{Max} \left(\frac{b_k}{\sum_{i=1}^{m} |b_i (B_{ki} - B_{kh})|} \right) \right].$$

"The maximum tolerance R^* is finite iff for some k an element of I - {n+1} there exists some i and j an element of {1,...,m} such that b_i not equal to zero and b_j not equal to zero and B_{ki} not equal to B_{kj}" (13:164).

2.5 Conclusion.

The search and rescue problem can be formulated as a maximum flow network and as a linear programming problem. The

network will consist of nodes representing each receiver station and each frequency. It will also have a supply node, a slack node, an excess coverage node, a non-excess coverage node, and a sink node. The frequency nodes are given a demand to ensure each frequency has a minimal coverage of at least two receivers. The arc costs of the excess coverage node are used to penalize the objective function for excess coverage of a frequency.

Sensitivity analysis is used to investigate the robustness of the frequency assignments produced by the network for different weights sequences. The network solution is used with the tolerance approach to produce an optimal range for the given weights of interest. The linear program solution is used to produce classical sensitivity analysis for the weight ranges.

III. Model Formulation

This chapter presents the multiobjective linear programming and network programming formulations of the search and rescue network. A literature review on this subject area suggests a mathematical model for this problem can be adapted from previous research by Johnson and Steppe (10;15). Both Johnson and Steppe showed a linear approximation of the actual non-linear network proved to be accurate in determining near-optimal frequency assignments and station locations. The only modification required is the addition of a weighting factor for the transmitter areas. Although several approaches exist to formulate this problem, linear programming and network programming are the techniques investigated in this study.

The weights for the transmitter areas are provided by the Department of Defense (DOD). The key factors in determining the different weight combinations are existing data bases and insight of DOD personnel. The next section describes the notation used in the linear formulation of this problem and the last section presents the entire multiobjective formulation.

3.1 Notation

The notation described in this section is used in the linear formulation of this problem for this study.

1. Subscripts.

   $\underline{i}$ = transmitter locations.

   $\underline{j}$ = receiving station locations.

   $\underline{k}$ = frequency bands.

2. Decision Variables.

$$X_{jk}=\begin{cases}1, & \textit{if station } j \textit{ is assigned cover frequency } k.\\ 0, & \textit{otherwise.}\end{cases}$$

$$Y_{k}=\begin{cases}n, & \textit{if frequency } k \textit{ has excess coverage by } n \textit{ stations.}\\ 0, & \textit{otherwise.}\end{cases}$$

3. Probabilities.

   $\underline{F}_{ik}$ = the probability of a distress signal from location $\underline{i}$ on frequency $\underline{k}$.

   $\underline{P}_{ijk}$ = the probability that a distress signal from location $\underline{i}$ on frequency $\underline{k}$ is acquired by station $\underline{j}$.

$\underline{W_{ij}}$ = the probability that a line of bearing from station $\underline{j}$ is within the acceptable circularized error region defined for location $\underline{i}$.

$\underline{U_i}$ = the normalized weight (0 - 1 range) of a distress signal from location $\underline{i}$.

4. Other.

TN = the total number of HFDF receivers.

FS = the fairshare of HFDF receivers for each frequency. Where FS is the integer greater than or equal to the total number of HFDF receivers divided by the total number of frequencies to be covered.

3.2 Objective Function 1. The first objective function adds the weight for area $\underline{i}$ to Steppe's first objective function (15:22). This objective function maximizes the expected number of accurate lines of bearing for HFDF receivers. The mathematical formula for the objective function is

$$Max \sum_{i} \sum_{j} \sum_{k} u_i w_{ij} F_{ik} P_{ijk} X_{jk}$$

3.3 Objective Function 2. The second objective function is identical to Johnson's third objective function (10:19). It minimizes the amount of excess coverage of HFDF receivers to each frequency. Excess coverage of a frequency is considered to be the number of HFDF receivers covering a frequency more than the "fairshare" number of HFDF receivers allowed to cover the frequency. The mathematical formula for this objective function is

$$Min \sum_{k} Y_k$$

which is identical to

$$Max \sum_{k} (-Y_k)$$

3.4 Constraints. The linear formulation of this problem contains three constraints. The first constraint limits the number of HFDF frequency assignments at each station to the number of receivers located at each station. The formulation

for this constraint is:

$$\sum_{k} X_{jk} \leq m_j \quad \forall j$$

The second constraint requires at least two HFDF receivers be assigned to cover every frequency. This constraint is needed because it takes three lines of bearing (LOB) to detect a distress signal. The third LOB comes from a RS system. The formulation for this constraint is:

$$\sum_{j} X_{jk} \geq 2 \quad \forall k$$

The third constraint determines the amount of excess coverage given to each frequency. The variable $\underline{Y}_k$ is the measure for excess coverage. The formulation for this constraint is:

$$\sum_{j} X_{jk} - Y_k \leq FS \quad \forall k$$

3.5 Complete Formulation. The complete linear model to be used in this study is:

$$Max \sum_i \sum_j \sum_k u_i w_{ij} F_{ik} P_{ijk} X_{jk}$$

$$Max \sum_k (-Y_k)$$

subject to

$$\sum_k X_{jk} \leq m_j \quad \forall j$$

$$\sum_j X_{jk} \geq 2 \quad \forall k$$

$$\sum_j X_{jk} - Y_k \leq FS \quad \forall k$$

$$X_{jk} = 0,1 \quad \forall j,k$$

$$Y_k \geq 0 \quad \forall k \ (k \ is \ integer)$$

This model is formulated and solved as a max flow network and as a linear program. The solutions are analyzed with sensitivity analysis to determine the robustness of the frequency assignments with regard to the weights associated with the transmitter areas. The next chapter describes the methodology used to solve and analyze this problem.

IV. Methodology

This chapter presents a brief description of the techniques used to solve and evaluate this problem. Linear programming is described in the first section. A general overview of network programming is provided in the next section. The third section covers multiobjective programming and includes a discussion about lambda values and scaling with regard to objective functions. The fourth section details a third method of sensitivity analysis used in this project. The next section describes a DOD software program used to evaluate the frequency assignments generated by the models. It also discusses a FORTRAN program created for this research project to evaluate the areas of interest. Finally, the last section presents the solution methodology used for this project.

4.1 Linear Programming.

Linear programming is a fairly recent discovery (conceived in 1947 by George Dantzig) which has met wide acceptance in many mathematical arenas because of "(1) its ability to model important and complex management decision problems and (2) its capability for producing solutions in a reasonable amount of time" (1:1). Linear programming uses a mathematical model

consisting of decision variables, an objective function, and constraints. The object is to find the optimal value of the objective function subject to the constraints of the problem. The optimal solution not only produces the maximum (or minimum) value of the objective function but it also yields the optimal values of the decision variables.

Linear programming models implicitly contain four assumptions for each problem (1:3). The first assumption is proportionality. The main idea of this assumption is if the level of a decision variable, say $\underline{x}_i$, is doubled, then its contribution to the objective function is also doubled. The second assumption is additivity. Additivity means the sum of the individual costs equals the total cost and the contribution of each constraint is the sum of the individual activities in that constraint. It also assures no interaction effects between the activities (1:3-4). The third assumption is divisibility. This means the decision variables may have any positive rational value, integer or fractional. The fourth and final assumption is called deterministic. This assumption means the objective function coefficients, constraint coefficients, and right-hand-side values for the constraints are known quantities. This means any randomness is assumed to be accounted for in the given values.

Although these assumptions are part of every model they do not necessarily hold completely true for every problem. The

reason for this is too much detail is required for large problems to ensure the assumptions are not violated. The exception being the divisibility assumption. This assumption holds for every linear (not integer) programming problem. The assumption that may be most violated is the deterministic assumption. This is why sensitivity analysis has gained so much importance in this field (8:33).

The simplex method developed by Dantzig is an algorithm for solving linear programs. The simplex method consists of three phases. The first phase is the initialization phase. This phase puts the problem into canonical form and finds an initial basic feasible solution. This is done by introducing slack variables into the constraints. These slack variables are the basic variables for the initial basic feasible solution. The second phase is the iterative phase. This phase determines the entering basic variable, the leaving basic variable, and the new basic feasible solution. The third and final phase is the optimality phase. This phase tests the current basic feasible solution to see if any feasible solutions are better. If no solution is better, then stop. If there are better solutions then go to the iterative phase. Continue until an optimal solution is found or until the problem is shown to be unbounded or infeasible (8:63-64).

4.2 Network Programming.

Network programming uses a modified version of the simplex method to generate optimal solutions. "This specialization, known as the network simplex algorithm, performs the simplex operations directly on the graph or the network itself" (1:419). The solution times for problems using the network simplex algorithm are several hundred times faster than solutions from the standard simplex algorithm (1:419). The network simplex algorithm accomplishes faster execution times by taking advantage of the special structure of the network problem. The algorithm still determines the entering basic variable, the leaving basic variable, and the feasible solution for each iteration but it does not generate any tableaux (8:379).

One of the concepts that the algorithm employs is an upper bounding technique. This technique provides constraints which limit the capacity for some or all of the arcs in the network. These constraints are treated like non-negativity constraints rather than functional constraints in the network algorithm. This means they are only used for determining the leaving basic variable at each iteration (8:379).

The second and most important concept of the network simplex algorithm is the correspondence between basic feasible solutions and feasible spanning trees (8:380). "A key property of basic arcs is that they never form undirected

cycles. Therefore, any set of basic arcs forms a spanning tree" (8:381). A feasible spanning tree is a spanning tree solution from the node constraints that also satisfies capacity and non-negativity constraints (8:381). By using these two main concepts the network simplex algorithm is able to solve many problems faster and easier than standard simplex operations.

4.3 Multiobjective Programming.

This research project is a multiobjective programming problem. It seeks to maximize the probability of detecting a distress signal while also minimizing the excess coverage provided to each frequency. The ADBASE program used on the problem in chapter four is not capable of optimizing a problem of this size (10). The problem is so large that ADBASE uses all the memory allocated to it before it can find the efficient points. Therefore, the two objective functions are combined into a single objective function using the lambda values determined in previous research and by scaling. The single objective problem is solved by linear and network programming described above.

The lambda values for the two objective functions were calculated by Johnson and Steppe. The lambda value for the first objective function is taken to be 0.91 and the lambda value for the second objective function is 0.09 (15). As

required, the lambda values sum to unity. Also, the magnitude of the second objective function is 100 times greater than the first objective function. Therefore, the second objective function is divided by one hundred in order to bring the two to the same magnitude.

4.4 Sensitivity Analysis.

In addition to the methods of classical sensitivity analysis and the tolerance approach described in the literature review, a third approach to sensitivity analysis is applied to this project. This third approach manually increments and decrements weights of particular areas to determine how large of a range exists for the weights in which the same solution remains optimal. This approach looks at five weights in time block one and six weights in time block 6. These weights are also used to examine how the optimal value of the project changes as the weights change. In addition, the results from a DOD program to evaluate the optimal frequency assignments are compared to the change in the weight values.

4.5 Assignment Evaluation.

EVAL is a FORTRAN computer program developed by the DOD. It evaluates the frequency assignments generated in the linear formulation of this project by computing the approximate

objective function value for the true non-linear network for the given assignments. EVAL has proven to be accurate to within five percent of the non-linear objective function while using $1/10^6$ of the processing time to solve the non-linear problem. In order to provide a basis for comparison, the DOD produced 5050 random frequency assignments for time block 6 and 10000 random assignments for time block 1. These random assignments were then input to EVAL to produce a mean, variance, and a standard deviation for the random assignments for each time block. The results of these random assignments and the results from this research project are compared in the chapter six.

Another method of evaluation comes from a FORTRAN program created for this project that computes the probability of detecting a distress signal for each of the forty areas. The probabilities for the areas of interest are computed for each weight sequence and these values are compared to similar probabilities computed by EVAL for each of the weight sequences.

4.6 Solution Methodology.

1. For time blocks 1 and 6, formulate problem as a network and solve for various weight sequences. This is a multiobjective problem with a single objective function of the form $Z = lambda_1*f_1+lambda_2*f_2$.

2. For time blocks 1 and 6, use the tolerance approach to determine the maximum range that the weights of interest may change while keeping the same optimal solution.

3. For time blocks 1 and 6, formulate as a linear program and solve for the initial weights given by the DOD and produce classical sensitivity analysis.

4. Input the optimal frequency assignments from above to EVAL and compare to the random assignments generated by the DOD.

5. Compare the probability of detection for the areas of interest for EVAL and the research project.

V. Sample Problem

In order to gain insight into and experience with this project, a sample problem from a subset of the entire project data was generated. This problem was formulated both as a network problem and as a linear program. The sample problem consists of four transmitter areas, five receiving stations, and three frequencies. Each model was solved nine different times, each time with a different weighting sequence for the four areas of interest. The first solution for each model had equal weights for the four areas. This solution was labelled the _a priori_ frequency assignment. The value of the _a priori_ frequency assignment solution value was then calculated for each of the other eight weight sequences. Also, the "true" optimal solutions were obtained by running the models for the eight remaining weight sequences. The _a priori_ and "true" solutions were then compared for the eight sequences to determine the percent difference in the two solutions.

Additionally, the problem was formulated as a multiobjective linear program. The multiobjective program was solved by two techniques. The first technique is the constraint reduced feasible region technique. The second technique is the multicriteria simplex method. A linear programming package was used for the first technique and a

software package called ADBASE was used for the second. The ADBASE program is used to determine all efficient extreme points for the problem and to produce a lambda cone for the problem. An efficient extreme point consists of values for each objective function. A point is an efficient extreme point when the value for one objective function cannot be increased without decreasing the value of another objective function. The lambda cone refers to the multipliers for each objective function whose sum equals unity and whose value each is limited to the interval from zero to unity.

5.1 Model Formulation

a. Linear Program Formulation. This problem was formulated as a linear program and was solved using SAS PROC LP. The objective function consists of two parts. The first part selects which frequencies are covered by the HFDFs at each station and the second part penalizes the excess coverage given to any frequency. The second part ensures a better balance of coverage for all frequencies. The A matrix for this problem is totally unimodular (TUM) which guarantees an integer solution.

All subscripts, decision variables, probabilities, and other notation are the same as indicated in chapter three. Also, the objective functions and constraints are exactly the same as in chapter three.

b. Max Flow Network Formulation. This problem was also formulated as a max flow network flow problem and was solved using SAS PROC NETFLOW. There are five nodes representing the receiving stations and three nodes representing the frequencies. The arcs between the receiving station nodes and the frequency nodes represent the frequency assignments. The capacity of these arcs is one and the costs of these arcs are the coefficients of the first part of the objective function from the linear programming formulation above. Excess and non-excess coverage are represented by two separate nodes. The costs of the arcs from the frequency nodes to the excess coverage node are the objective function coefficients from the second part of the linear formulation and have a capacity of infinity. The cost of the arcs from the frequency nodes to the non-excess coverage node is zero because there is no cost for non-excess coverage. Also, the capacity of these arcs is 3 because more than 3 frequency assignments for the same frequency would cause an excess in coverage. There is also a slack node that is assigned all HFDFs not assigned to the receiving station nodes.

c. Multiobjective Formulation. "Pareto preference is based on the concept of more is better for each criterion f_i, $\underline{i}=1, \ldots, q$; and no other information about the tradeoff of $\{f_i\}$ is established or available" (17:21). In Pareto

optimality, the N-points with respect to Pareto preference are the Pareto optimal solutions. N-points are the set of nondominated solutions for the problem. N-points are also called efficient points, noninferior points, nondominated points, or admissible points (17:22). The alternative set X consists of the same constraints listed previously and the criteria set f consists of the two objective functions already listed above.

d. Constraint Reduced Feasible Region Formulation. In order to find the N-points for this multiobjective problem, the problem was converted into a single-criterion mathematical programming problem. The single criterion is the first criterion, $f_1(X)$. The second criterion, $f_2(X)$, becomes a greater-than-or-equal to constraint with a right-hand-side of r_2, where r_2 is the satisficing level for $f_2(X)$. The new constraint is maximized and minimized over the X space to yield a range of right-hand-side values for the constraint. The linear program is then solved with r_2 varying from the minimum value to the maximum value. The solutions to the linear program are the N-points in the X-space. These points are then used in the criterion functions to find the N-points in the Y-space. The problem looks like this:

$$\max f_1(X) = 0.0043x_{11} + 0.004275x_{12} + 0.004725x_{13}$$
$$+ 0.007325x_{21} + 0.00726x_{22} + 0.00736x_{23}$$
$$+ 0.001938x_{31} + 0.0032x_{32} + 0.002725x_{33}$$
$$+ 0.001183x_{41} + 0.0013x_{42} + 0.001103x_{43}$$
$$+ 0.007813x_{51} + 0.007495x_{52} + 0.0076x_{53}$$

subject to:

$$f_2(X) = - y_1 - y_2 - y_3 >= r_2$$
$$x_{11} + x_{12} + x_{13} + x_{21} + x_{22} + x_{23} + x_{31} + x_{32} + x_{33}$$
$$+ x_{41} + x_{42} + x_{43} + x_{51} + x_{52} + x_{53} <= 15$$
$$x_{11} + x_{21} + x_{31} + x_{41} + x_{51} - e_1 <= 3$$
$$x_{12} + x_{22} + x_{32} + x_{42} + x_{52} - e_2 <= 3$$
$$x_{13} + x_{23} + x_{33} + x_{43} + x_{53} - e_3 <= 3$$
$$x_{11} + x_{21} + x_{31} + x_{41} + x_{51} >= 2$$
$$x_{12} + x_{22} + x_{32} + x_{42} + x_{52} >= 2$$
$$x_{13} + x_{23} + x_{33} + x_{43} + x_{53} >= 2$$

e. Multicriteria Simplex Formulation. The Multicriteria Simplex (MC Simplex) algorithm is a generalized form of the single-criteria simplex algorithm. It is used to solve linear problems with more than one objective function. The procedure for performing the algorithm is as follows: (1) set up a tableau just as in the standard simplex method except that the $z_j - c_j$ row is actually numerous $z_j - c_j$ rows, one for each objective function; (2) variables are pivoted into the basis based on revised regular simplex rules to include dominance of

variables; (3) an N-point is achieved when a variable cannot be introduced into the basis without decreasing the value of one of the objective functions; (4) all N-points have been found when the addition of a new variable to the basis yields the tableau for the first N-point that was found.

The problem for MC Simplex is:

$$\begin{aligned}\max f_1(X) = {} & 0.0043x_{11} + 0.004275x_{12} + 0.004725x_{13} \\ & + 0.007325x_{21} + 0.00726x_{22} + 0.00736x_{23} \\ & + 0.001938x_{31} + 0.0032x_{32} + 0.002725x_{33} \\ & + 0.001183x_{41} + 0.0013x_{42} + 0.001103x_{43} \\ & + 0.007813x_{51} + 0.007495x_{52} + 0.0076x_{53}\end{aligned}$$

$$\max f_2(X) = - y_1 - y_2 - y_3$$

subject to:

$$\begin{aligned}& x_{11} + x_{12} + x_{13} + x_{21} + x_{22} + x_{23} + x_{31} + x_{32} + x_{33} \\ & \quad + x_{41} + x_{42} + x_{43} + x_{51} + x_{52} + x_{53} <= 15 \\ & x_{11} + x_{21} + x_{31} + x_{41} + x_{51} - e_1 <= 3 \\ & x_{12} + x_{22} + x_{32} + x_{42} + x_{52} - e_2 <= 3 \\ & x_{13} + x_{23} + x_{33} + x_{43} + x_{53} - e_3 <= 3 \\ & x_{11} + x_{21} + x_{31} + x_{41} + x_{51} >= 2 \\ & x_{12} + x_{22} + x_{32} + x_{42} + x_{52} >= 2 \\ & x_{13} + x_{23} + x_{33} + x_{43} + x_{53} >= 2\end{aligned}$$

5.2. Data

Table 1 Probability of a Signal from Transmitter i on Frequency k.

i/k	Frequency 1	Frequency 2	Frequency 3
Trans 1	.04	.04	.04
Trans 2	.00	.00	.01
Trans 3	.03	.05	.05
Trans 4	.00	.00	.00

Table 2 Probability of a Signal from Transmitter i on Frequency k is Acquired by Station j.

j	Transmitter				Transmitter				Transmitter			
	1	2	3	4	1	2	3	4	1	2	3	4
1	98	32	51	01	95	13	35	01	96	33	52	01
2	98	44	13	01	98	08	01	01	98	30	01	01
3	97	01	01	01	92	46	71	01	83	31	51	01
4	97	97	01	01	98	01	12	01	90	01	01	01
5	98	29	01	01	94	04	01	01	94	19	00	01

Table 3 Probability Station j will Receive a Signal from Transmitter i Given that a Signal was Transmitted.

i/j	Station 1	Station 2	Station 3	Station 4	Station 5
Trans 1	.3808	.7407	.1951	.1210	.7956
Trans 2	.1477	.1301	.1140	.0596	.2504
Trans 3	.1471	.0892	.1580	.0834	.1509
Trans 4	.0515	.7679	.0615	.0820	.0427

Table 4 Weighting Sequences for the Nine Solutions to the Sample Problem (i represents the four areas of interest and s represents the nine sequences).

i/s	1	2	3	4	5	6	7	8	9
1	.25	.50	.167	.167	.167	.70	.10	.10	.10
2	.25	.167	.50	.167	.167	.10	.70	.10	.10
3	.25	.167	.167	.50	.167	.10	.10	.70	.10
4	.25	.167	.167	.167	.50	.10	.10	.10	.70

5.3. Problem Solution

a. Linear Formulation. This problem was solved nine times using linear programming with different weighting sequences. The first solution was for all the u_is set to 0.25. This solution's assignment of frequencies is the *a priori* solution and will be used to find the *a priori* solution for the other eight weighting sequences. Only two of the weight sequences resulted in different values for the optimal and *a priori* solutions.

b. Network Formulation. This problem was solved for the same weighting sequences used in the linear problem above. In addition, the problem was solved manually for the nine weighting sequences using the *a priori* solution of the original problem. The *a priori* solution and the true optimal solution are then compared to determine the percent difference in the two solutions. The solutions are shown in Appendix A.

c. Constraint Reduced Solution. Minimizing and maximizing $f_2(X)$ over the X space yields a minimum value of r_2 = -6 and a maximum value of r_2 = 0. This problem was solved using SAS PROC LP for the range of r_2 listed above. The following N-points were obtained:

	X space N-points	Y space N-points	
r_2	$(x_{11}, \ldots, x_{53}, e_1, e_2, e_3)$	$(f_1(X), f_2(X))$	
-6	(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,2,2)	(0.069602, -6)	A
-5	(1,1,1,1,1,1,1,1,1,1,1,0,1,1,1,2,2,1)	(0.068499, -5)	B
-4	(1,1,1,1,1,1,1,1,1,0,1,0,1,1,1,1,2,1)	(0.067316, -4)	C
-3	(1,1,1,1,1,1,1,1,1,0,0,0,1,1,1,1,1,1)	(0.066016, -3)	D
-2	(1,1,1,1,1,1,0,1,1,0,0,0,1,1,1,0,1,1)	(0.064078, -2)	E
-1	(1,1,1,1,1,1,0,1,0,0,0,0,1,1,1,0,1,0)	(0.061353, -1)	F
0	(1,1,1,1,1,1,0,0,0,0,0,0,1,1,1,0,0,0)	(0.058153, 0)	G

d. MC-Simplex Solution. ADBASE is a software package that uses MC Simplex to find the N-points and the lambda cone for multicriteria problems. ADBASE is FORTRAN-based and it requires two input files to operate correctly. The first file has an extension of .QFI. The aaa.QFI file contains settings for the different phases and aspects of ADBASE. It also contains settings for what type of output is printed. The

.QFI file is divided into two sections. The first section is for mode 1 and the second section is for mode 2. Both sections must be completed for every problem, even if the problem is only using a single mode, or the ADBASE software will generate a FORTRAN READ ERROR. The second file has an extension of .IFI. The aaa.IFI file contains the data ADBASE uses. It contains the objective function coefficients, the constraint coefficients, the constraint right-hand-side values, and the lambda ranges for the objective functions. The .IFI file has specific fields that data must be in or the ADBASE software will not work properly.

The following N-points in the X space and the Y space were found by ADBASE:

r_2	X space N-points $(x_{11}, \ldots, x_{53}, e_1, e_2, e_3)$	Y space N-points $(f_1(X), f_2(X))$	
-6	(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,2,2,2)	(0.069602, -6)	A
-5	(1,1,1,1,1,1,1,1,1,1,1,0,1,1,1,2,2,1)	(0.068499, -5)	B
-4	(1,1,1,1,1,1,1,1,1,0,1,0,1,1,1,1,2,1)	(0.067316, -4)	C
-3	(1,1,1,1,1,1,1,1,1,0,0,0,1,1,1,1,1,1)	(0.066016, -3)	D
-2	(1,1,1,1,1,1,0,1,1,0,0,0,1,1,1,0,1,1)	(0.064078, -2)	E
-1	1,1,1,1,1,1,0,1,0,0,0,0,1,1,1,0,1,0)	(0.061353, -1)	F
0	(1,1,1,1,1,1,0,0,0,0,0,0,1,1,1,0,0,0)	(0.058153, 0)	G

These are the same N-points as were found in the previous section. Therefore, the graphs of the X space and the Y space are the same as above as well.

e. Lambda Cone for f_1 and f_2. The lambda cone for this problem was calculated by ADBASE. The point G is the maximum point for $(\text{lambda}_1, \text{lambda}_2) = (0,1)$ to approximately $(0.997,0.003)$. The point F is the maximum point for $(\text{lambda}_1, \text{lambda}_2) = (0.997,0.003)$ to $(0.998,0.002)$. The point E is the maximum point for $(\text{lambda}_1, \text{lambda}_2) = (0.998,0.002)$ to $(0.9985,0.0015)$. The point D is the maximum point for $(\text{lambda}_1, \text{lambda}_2) = (0.9985,0.0015)$ to $(0.9987,0.0013)$. The point C is the maximum point for $(\text{lambda}_1, \text{lambda}_2) = (0.9987,0.0013)$ to $(0.9988,0.0012)$. The point B is the maximum point for $(\text{lambda}_1, \text{lambda}_2) = (0.9988,0.0012)$ to $(0.9989,0.0011)$. The point A is the maximum point for $(\text{lambda}_1, \text{lambda}_2) = (0.9989,0.0011)$ to $(1,0)$.

5.4. Conclusion

This problem was formulated as a linear programming problem, a max flow network problem, and a multiobjective problem. Both the linear and network formulations gave the same values for all nine weighting sequences. The sample problem solutions have shown that the *a priori* solution was at most 7% less than the optimal solution and at best the same as the optimal solution. This indicates the optimal solutions

form a plateau in the Y-space void of any sharp peaks and that the solution for any weight sequence of this problem will lie on the plateau. Because this is only a very small subset of the real problem, the results and conclusions obtained in this chapter cannot be used to predict the behavior of the real problem.

This problem, when formulated to find Pareto-optimal solutions, is shown to have seven Pareto-optimal points. It was not difficult to determine the N-points for this problem because the range for r_2 was small, as was the size of the problem.

When ADBASE was used to formulate and solve this problem it produced the same seven N-points as determined by SAS. It was also used to produce the lambda cone for this problem. Although ADBASE is not user-friendly, it is much more convenient to use than either of the SAS programs. It also makes it much easier to compute the lambda cone.

The next chapter presents the results for the entire research problem. It also draws some conclusions from the results and makes some recommendations for future research.

VI Results, Conclusions, and Recommendations

This chapter reports the results of the solution methodology listed in the chapter four. The first section compares how the optimal value from the network and the EVAL program change as the weights increase from zero to ten times the magnitude from the original DOD weight sequence. The second section evaluates the probability of geolocation of each area of interest from the project and EVAL with the weight sequences. The real measures of merit in the first two sections are the values from the EVAL program. The EVAL values represent what is happening in the true nonlinear network whereas the linear network value has no real meaning to the nonlinear problem. The next section presents the optimal weight ranges for the weights of interest for the three sensitivity analysis approaches. The next section compares the network solutions to the random assignment EVAL solutions. The fifth section draws some conclusions from these results and the last section presents some recommendations for further research.

6.1 Solution Comparisons.

This section describes how the solutions from the linear network and the EVAL program change as the weights of interest

change. Each weight is described by two graphs. The first graph compares the change in the linear network and the change in the EVAL program. The second graph shows how the measure of geolocation for each weight area of interest changes as the weight changes. This section is divided into two sections. The first section is for time block one and the second for time block six.

6.1.1 Time Block 1. The five weights of interest for time block one are number 20, 22, 27, 30, and 31. The weight ranges for these weights are listed in Appendix A. Weights 20, 27, and 31 range in value from 0.0 to 0.717949 and weights 22 and 30 range from 0.0 to 0.628931. Weights 20, 22, 30, and 31 have fifteen different values to be evaluated. Weight 27 has sixteen values to be evaluated.

The first weight to be examined is weight number 20. The linear network was solved for each weight value and EVAL produced a numerical evaluation of each optimal frequency assignment for the true non-linear network. The optimal linear network solutions and the EVAL solutions are displayed in Figure 1 and the network area of interest geolocation values and the EVAL area of interest geolocation values are shown in Figure 2.

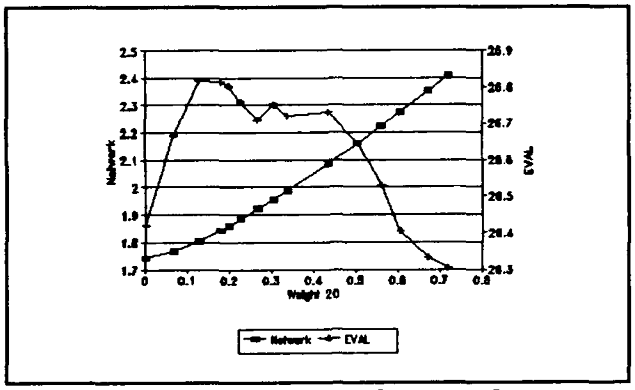

Figure 1 Network and EVAL Overall Solutions for Weight 20, Time Block 1

Figure 1 shows that weight 20 and the network solution have a linear relationship where the network value increases as the weight value increases. On the other hand, the EVAL solution has a non-linear relationship with weight 20. The EVAL solution increases for very small values of weight 20 but it decreases for values over 0.1.

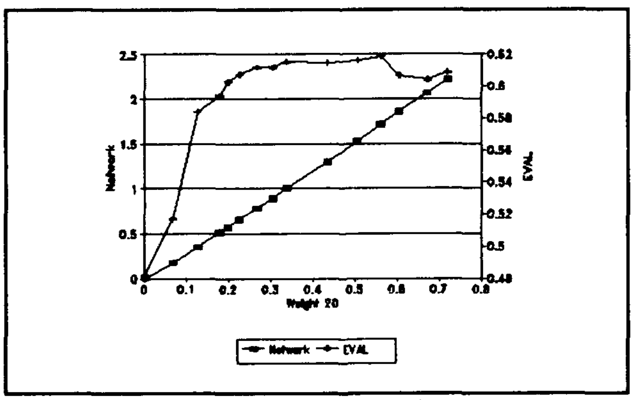

Figure 2 Network and EVAL Area of Interest Detection Values for Weight 20, Time Block 1

Figure 2 indicates a linear relationship for weight 20 and the geolocation value for area 20 in the linear network. It also shows a nonlinear relationship between the weight value and the EVAL geolocation value for area 20. The EVAL geolocation value increases faster for small values of the weight than for large values. These two figures show that larc̜e values of weight 20 will increase the geolocation value for area 20 but causes a decrease in the overall geolocation value for the non-linear network.

The next weight to be discussed is weight number 22. Like weight 20, the first graph shows the overall solutions and the second shows the area geolocation values.

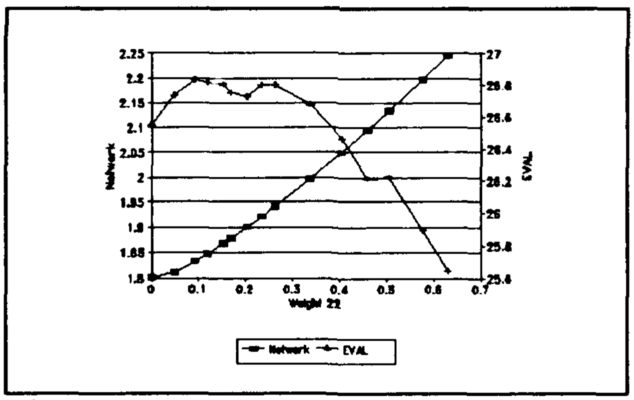

Figure 3 Network and EVAL Overall Solutions for Weight 22, Time Block 1

Figure 3 shows that weight 22 and the network solution, like weight 20, are linearly related with the

network value increasing as the weight value increases. But, the EVAL solution decreases with weight values over 0.1.

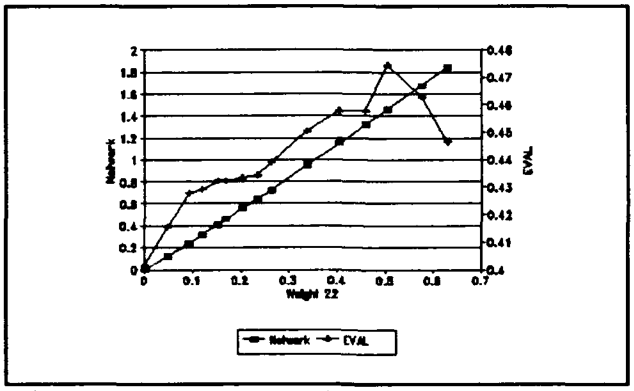

Figure 4 Network and EVAL Area of Interest Detection Values for Weight 22, Time Block 1

This figure indicates a linear relationship for weight 22 and the geolocation value for area 22 in the network. It also shows a non-linear relationship between the weight value and the EVAL detection value for area 22. The EVAL geolocation value increases slowly for weight values between 0.0 and 0.49 and then decreases for values over 0.49. These two figures show that moderate values of weight 22 produce the largest geolocation values for area 22 but any value over 0.1 causes a decrease in the overall geolocation value for the non-linear network.

The third weight to be discussed is weight number 27. Weight 27 was given sixteen different values for which the

network and EVAL solutions were obtained. Figures 5 and 6 describe the overall and area geolocation value relationships with weight 27.

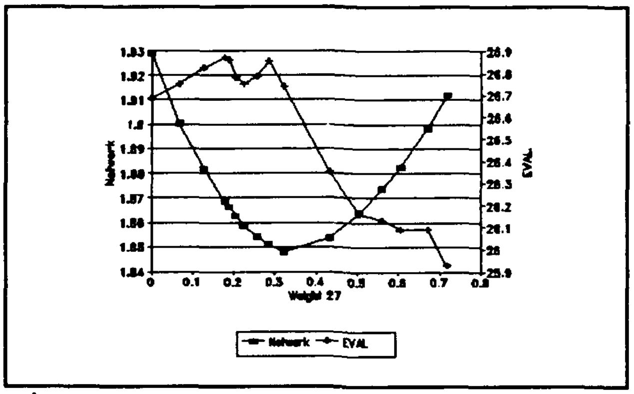

Figure 5 Network and EVAL Overall Solutions for Weight 27, Time Block 1

The figure shows a nonlinear relationship for weight 27 and the network solution. But, again, the EVAL solution varies over the weight values. EVAL increases as the weight increases from 0.0 to 0.2 but then decreases over the range 0.29 to 0.72.

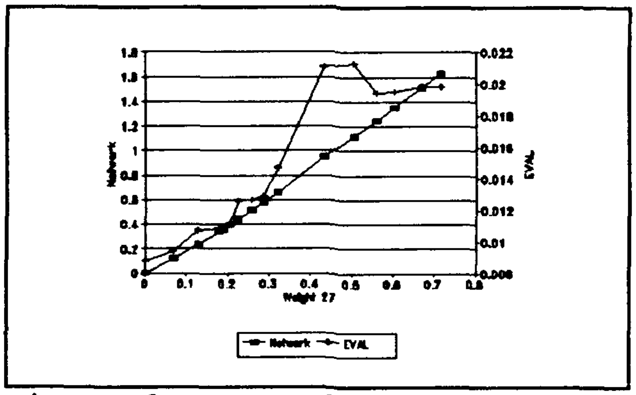

Figure 6 Network and EVAL Area of Interest Detection Values for Weight 27, Time Block 1

Figure 6 shows a nonlinear relationship between weight 27 and the EVAL geolocation value for area 27. The EVAL geolocation value steadily increases slowly for weight values between 0.0 and 0.49 and then levels off. These two figures for weight 27 again show that large values of weight 27 produce the largest geolocation values for area 27 but also cause a decrease in the overall geolocation value for the nonlinear network.

The fourth weight to be discussed for time block one is weight number 30. Weight 30 was varied for fifteen different values for which the network and EVAL solutions were obtained. The relationships between the solutions and the weight are shown in the next two figures.

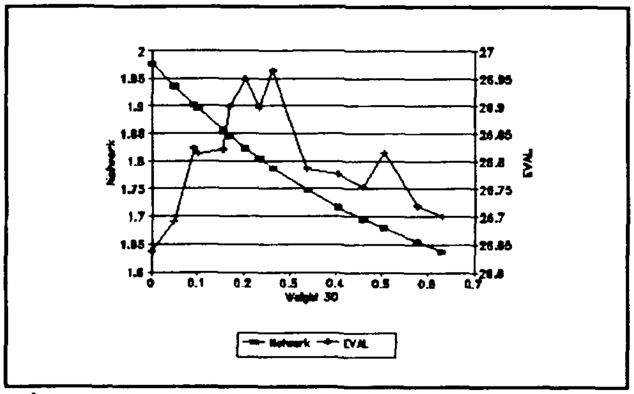

Figure 7 Network and EVAL Overall Solutions for Weight 30, Time Block 1

Figure 7 shows a linear relationship between weight 30 and the total network value but it is an inverse relationship. As

the weight increases in value, the network decreases in value. The EVAL relationship to the weight is also non-linear. The EVAL solution increases as the weight increases from 0.0 to 0.25 and then decreases for larger values of weight 30.

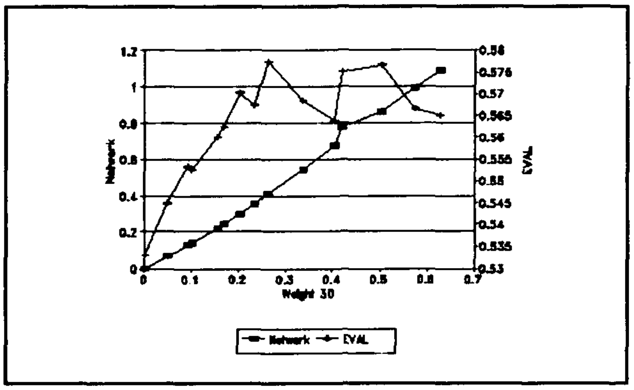

Figure 8 Network and EVAL Area of Interest Detection Values for Weight 30, Time Block 1

Figure 8 also shows that a linear relationship exists between the network area geolocation value and the weight value. It also shows a nonlinear relationship for the EVAL area geolocation value and the weight value. The geolocation value increases for weight 30 values between 0.0 and 0.27 and oscillates for weight values over 0.27. These two figures indicate a moderate value for weight 30, somewhere close to 0.27, produce the highest overall and area values for EVAL.

The fifth and final weight to be discussed for time block one is weight number 31. Figures 9 and 10 show the relationships between the network and EVAL solutions and weight 31.

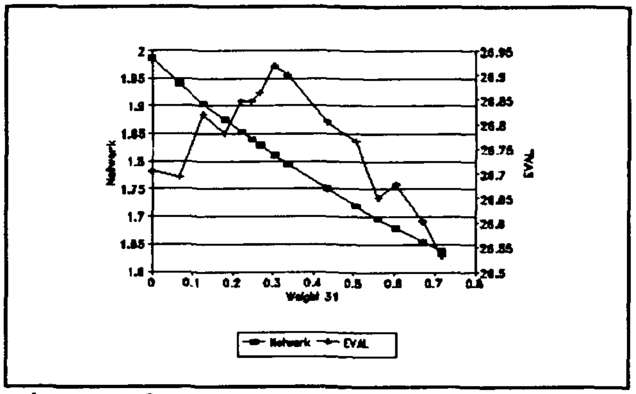

Figure 9 Network and EVAL Overall Solutions for Weight 31, Time Block 1

This figure shows the same inverse linear relationship between the network solution and the weight value as Figure 7. And again, a nonlinear relationship exists between the EVAL solution and weight 31. The EVAL solution increases for weight 31 values below 0.3 and decreases for values over 0.3.

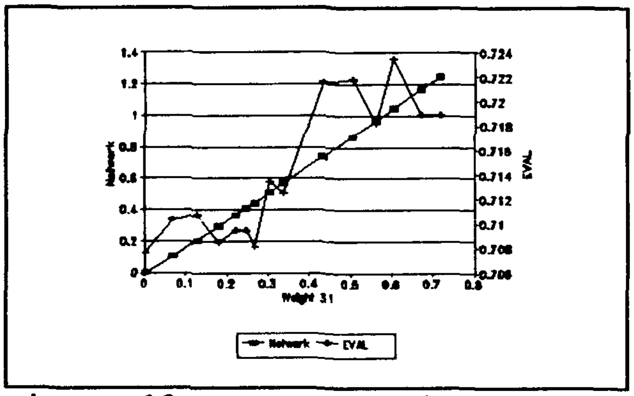

Figure 10 Network and EVAL Area of Interest Detection Values for Weight 31, Time Block 1

Figure 10 shows an oscillating nonlinear relationship between the weight value and the EVAL area geolocation value. The only range where the EVAL area 31 value always increases is between 0.27 and 0.49. Values below and above this range have oscillating values for area 31. These figures show that in order to increase area 31 geolocation value the weight value must increase but the overall geolocation value for EVAL will decrease. Therefore, a trade-off exists between the overall geolocation value and the area geolocation value for all five weights examined in this section.

<u>6.1.2 Time Block 6.</u> There are six weights of interest for time block six. They are number 9, 20, 27, 30, 31, and 40. The weight range for weights 9, 20, and 27 is from 0.0 to 0.636574. The range for weights 30 and 40 is 0.0 to 0.610501 and weight range for weight 31 is from 0.0 to 0.700701. All six weights have fifteen different values to be evaluated.

Weight number 9 is examined first. The linear network solution for each weight value and the EVAL evaluation of each optimal frequency assignment for the true nonlinear network are plotted on two graphs. The optimal linear network solutions and the EVAL solutions are displayed in Figure 11 and the network area of interest geolocation values and the EVAL area of interest geolocation values are shown in Figure 12.

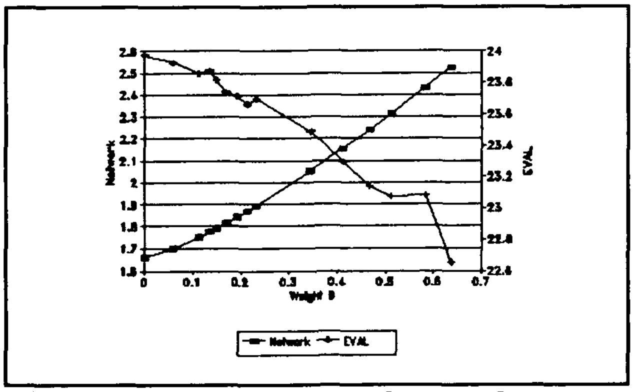

Figure 11 Network and EVAL Overall Solutions for Weight 9, Time Block 6

This figure shows that weight 9 and the network solution have a linear relationship where the network value increases as the weight value increases. The EVAL solution has a near inverse linear relationship with weight 9. The EVAL solution decreases for as the weight value increases.

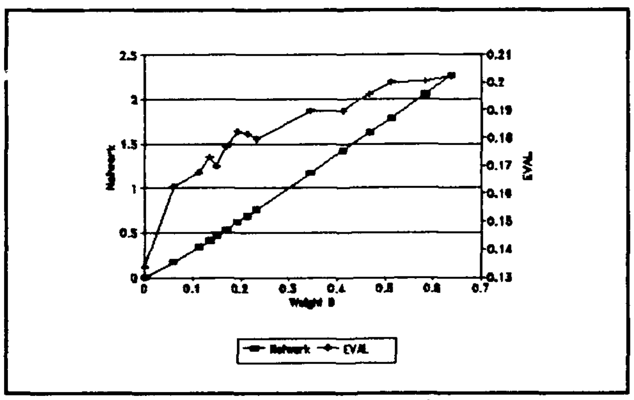

Figure 12 Network and EVAL Area of Interest Detection Values for Weight 9, Time Block 6

Figure 12 indicates a linear relationship exists between weight 9 and the geolocation value for area 9 in the linear network. It also shows a nonlinear relationship between the weight value and the EVAL geolocation value for area 9. The EVAL geolocation value increases faster for small values of the weight than for large values. These two figures show that large values of weight 20 increases the geolocation value for area 9 but causes a decrease in the overall geolocation value for the nonlinear network.

The next weight to be evaluated is weight number 20. As with weight 9, the first graph shows the overall solutions and the second shows the area geolocation values.

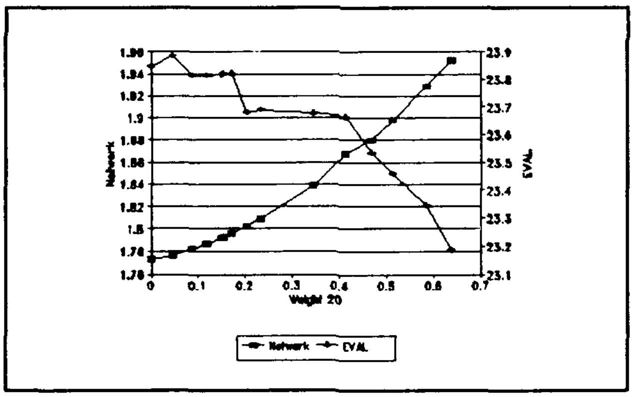

Figure 13 Network and EVAL Overall Solutions for Weight 20, Time Block 6

Figure 13 show basically the same relationships for the network solution and EVAL solutions with the weight value.

The network value increases as the weight value increases and the EVAL solution decreases as the weight increases.

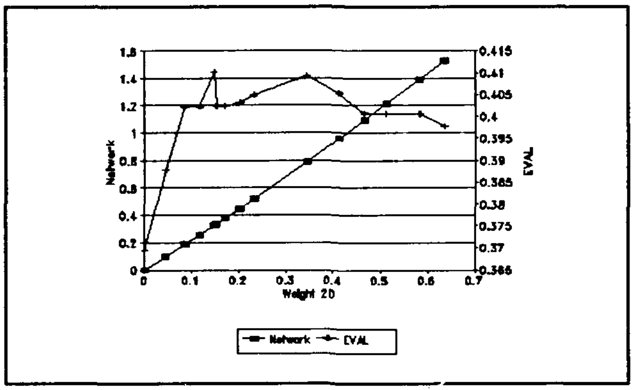

Figure 14 Network and EVAL Area of Interest Detection Values for Weight 20, Time Block 6

This figure indicates a linear relationship for weight 20 and the network geolocation value for area 20. It also shows a non-linear relationship between the weight value and the EVAL geolocation value for area 20. The EVAL geolocation value increases quickly for weight values between 0.0 and 0.13 and then decreases for values over 0.32. These two figures show that small values of weight 20 produce the largest overall solutions and geolocation values for area 20 for the nonlinear network.

The third weight to be discussed is weight number 27. Weight 27 was given fifteen different values for which the network and EVAL solutions were obtained. Figures 15 and 16

describe the overall and area geolocation value relationships with weight 27.

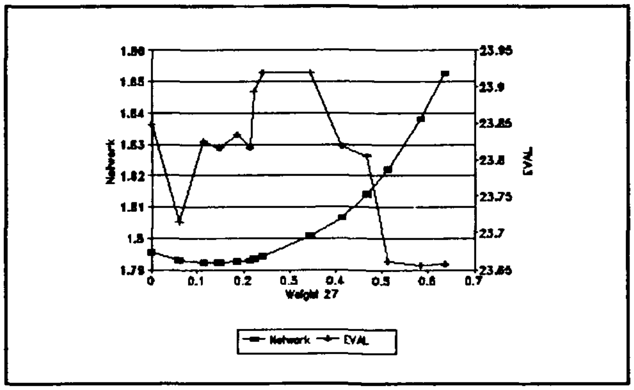

Figure 15 Network and EVAL Overall Solutions for Weight 27, Time Block 6

Figure 15 shows the same nonlinear relationship for weight 27 and the network solution as Figure 5 did for time block one. But, once again, the EVAL solution varies over the weight values. EVAL decreases as the weight increases from 0.0 to 0.07, increases over the range 0.07 to 0.35, and then decreases for values over 0.35.

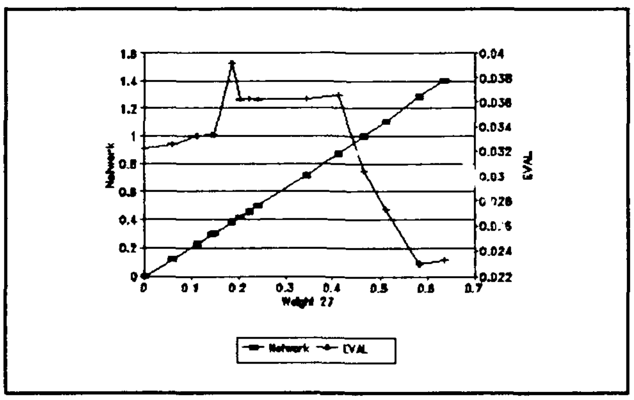

Figure 16 Network and EVAL Area of Interest Detection Values for Weight 27, Time Block 6

Figure 16 shows a nonlinear relationship between weight 27 and the EVAL geolocation value for area 27. The EVAL geolocation value increases for weight values between 0.0 and 0.19, is relatively flat between 0.19 and 0.41, and decreases sharply for weight values larger than 0.41. These two figures for weight 27 show that large values of weight 27 produce the smallest geolocation values for area 27 and the smallest overall geolocation value for the nonlinear network. The best range for both values is 0.22 and 0.36.

The fourth weight to be discussed for time block six is weight number 30. Weight 30 was varied for fifteen different values for which the network and EVAL solutions were obtained. The relationships between the solutions and the weight are shown in the next two figures.

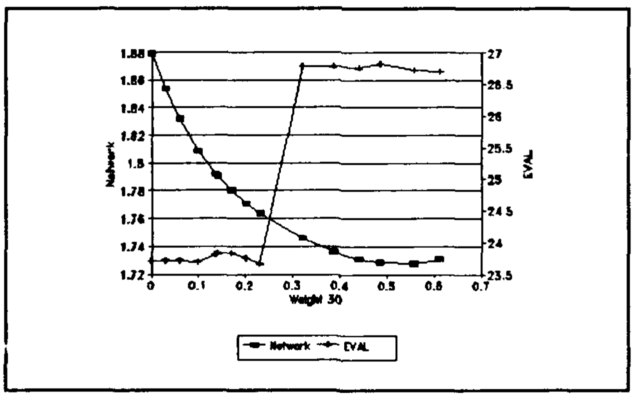

Figure 17 Network and EVAL Overall Solutions for Weight 30, Time Block 6

Figure 17 shows a nonlinear relationship between weight 30 and the total network value where the network value decreases as the weight value increases. The EVAL relationship to the weight is also nonlinear. The EVAL solution is mainly flat for the ranges 0.0 to 0.22 and 0.29 and 0.6 but it increases drastically for the weight range from 0.22 to 0.29.

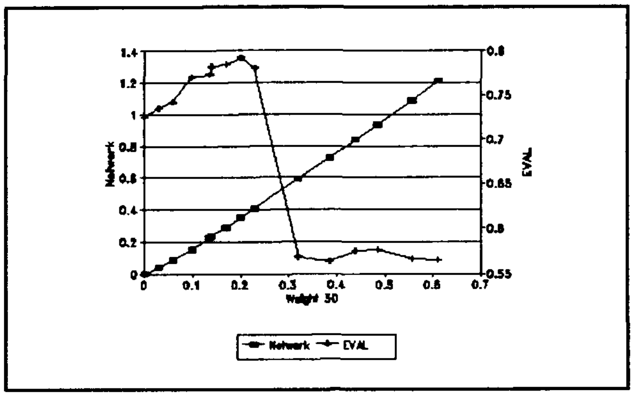

Figure 18 Network and EVAL Area of Interest Detection Values for Weight 30, Time Block 6

Figure 18 also shows that a linear relationship exists between the network area geolocation value and the weight value. It also shows a nonlinear relationship for the EVAL area geolocation value and the weight value. The geolocation value increases for weight 30 values between 0.0 and 0.21 and then decreases rapidly for weights between 0.21 and 0.32. These two figures indicate that the total EVAL solution and

the area EVAL solution are opposites of each other. As one increases, the other decreases, sometimes very quickly.

The fifth weight to be discussed for this time block is weight number 31. Figures 19 and 20 show the relationships between the network and EVAL solutions and weight 31.

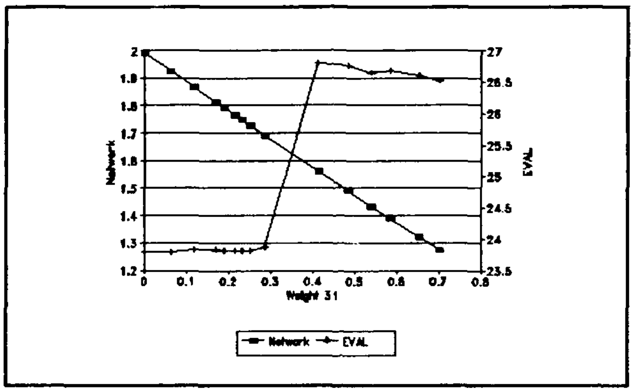

Figure 19 Network and EVAL Overall Solutions for Weight 31, Time Block 6

This figure shows the same nonlinear relationship exists between the EVAL solution and weight value as Figure 17. The EVAL solution is flat for values between 0.0 and 0.29 and values over 0.36 and it increases rapidly for weight values between 0.29 and 0.36.

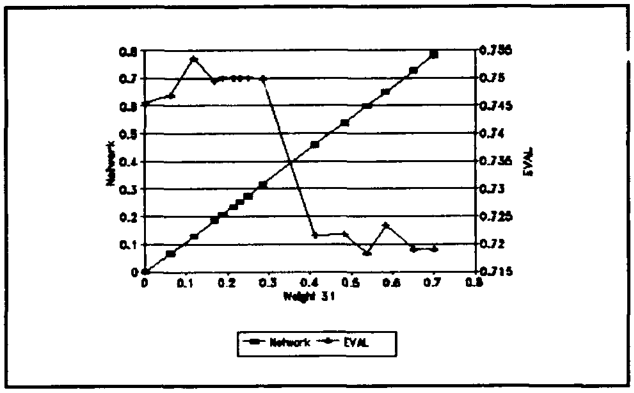

Figure 20 Network and EVAL Area of Interest Detection Values for Weight 31, Time Block 6

Figure 20 shows an oscillating nonlinear relationship between the weight value and the EVAL area geolocation value for values between 0.0 and 0.15 and values larger than 0.49. The range 0.29 to 0.41 shows a rapid decrease in the EVAL area 31 geolocation value. These figures again show that in order to increase the area 31 geolocation value the weight value must be small but the overall geolocation value for EVAL will decrease. Therefore, a trade-off exists between the overall geolocation value and the area geolocation value for the five weights examined so far in this section.

The sixth and final weight to be evaluated for time block six is weight 40. Figures 21 and 22 represent the relationships for weight 40 and the total solutions and the area geolocation values.

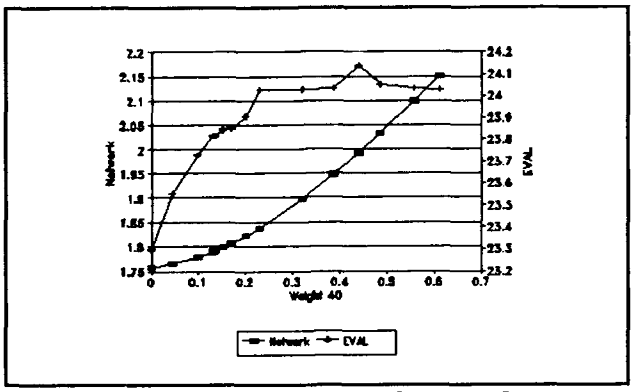

Figure 21 Network and EVAL Overall Solutions for Weight 40, Time Block 6

This figure shows a linear relationship for the network solution and a nonlinear relationship for the EVAL solution. The EVAL solution increases over the entire weight range investigated but smaller values for weight 40 produced larger increases than larger values for weight 40.

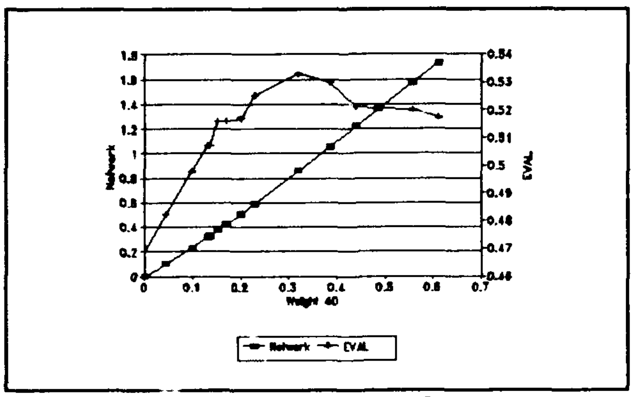

Figure 22 Network and EVAL Area of Interest Detection Values for Weight 40, Time Block 6

Figure 22 shows a nonlinear relationship for the EVAL area geolocation values and weight 40. The geolocation value increases as weight 40 increases from 0.0 to 0.31 and then decreases as weight 40 increases over 0.31. The network relationship is linear with weight 40 where the network value increases as the weight increases. These two figures show the largest weight range for any of the weights investigated for which both the EVAL overall solution and the EVAL area geolocation value increase as the weight value increases.

6.2 Sensitivity Analysis

This section describes three approaches used to determine the range for each weight of interest where the optimal solution remains unchanged. The range is calculated from the original weight sequences for time blocks one and six supplied by the DOD. The first approach is the tolerance approach to sensitivity analysis which looks at simultaneous independent variations in the cost coefficients. The second approach uses traditional sensitivity analysis generated by the linear formulation of the problem to calculate the weight ranges. The third and final approach is a manual change in the weight of interest. This approach re-solved the network for small increases and decreases in the weight values and determined where the optimal solution changed frequency assignments.

6.2.1 Tolerance Approach. The tolerance approach was performed on the network solutions for the original weight sequences for time block one and six. It was also performed n the optimal solutions for time blocks one and six where the weights for all forty areas are equal. The ranges for each solution in time block one and time block six compared to determine which produces the most robust frequency assignment.

6.2.1.1 Time Block One. The original weight sequence for time block one contained weights 20, 27, and 31 with a value of 0.203 and weights 22 and 30 with a value of 0.145. The other weights have a value of 0.003. The sum of the forty weights is unity. The percent change in the weights are listed in Table 5 below.

Table 5 Range for Original Weights in Time Block 1

Weight #	Original Value	Percent Change
20	.203	.00
22	.145	.00
27	.203	.00
30	.145	.00
31	.203	.00

The reason for the zero percent change in the original value is the reduced cost of one of the non-basic variables is zero. The reduced cost of the non-basic variables is the

numerator in the equation used to calculate the minimum percent change allowed. Therefore, if the reduced cost is zero, the percent change is zero.

The network was also solved for all forty weights equal to 0.025. The tolerance approach ranges for the original five weights are listed in table 6. The equal weights were used because of the results obtained using equal weights in the sample problem in chapter 4.

Table 6 Range for Equal Weights in Time Block 1

Weight #	Original Value	Percent Change
20	.025	.0128
22	.025	.0060
27	.025	.0040
30	.025	.0112
31	.025	.0024

The numbers in Table 6 show that weight 20 has the largest optimality range, followed by weight 30, weight 22, weight 27, and finally, weight 31. Because the original weights have no optimality range, the equal weights are more robust in maintaining the same optimal solution.

<u>6.2.1.2 Time Block Six.</u> The original weight sequence for time block six consisted of weights 9, 20, and 22 having a value of 0.1491, weights 30 and 40 have a value of 0.1355,

and weight 31 has a value of 0.1897. All other weights have a value of 0.0027. The sum of these weights is unity. The optimal ranges for the original six weights of interest are listed in Table 7.

Table 7 Range for Original Weights in Time Block 6

Weight #	Original Value	Percent Change
9	.1491	.00054
20	.1491	.00094
27	.1491	.00087
30	.1355	.00052
31	.1897	.00100
40	.1355	.00111

The results shown in this table indicate that weight 40 has the largest optimality range and *weight 30 has the* smallest. The percentages are very small for the tolerance approach and indicate that the weights given are not very robust.

The network was also solved for all forty weights equal to 0.025. The tolerance approach ranges for the same six weights are listed in table 8. The equal weights were used because of the results obtained using equal weights in the sample problem in chapter 4.

Table 8 Range for Equal Weights in Time Block 6

Weight #	Original Value	Percent Change
9	.025	.0040
20	.025	.0032
27	.025	.0108
30	.025	.0048
31	.025	.0020
40	.025	.0244

Table 8 shows percentages that are still relatively small but all are greater than the percentages for the original weight values. The percentage are eight times greater for weight 9, four times greater for weight 20, twelve times greater for weight 27, nine times greater for weight 30, two times greater for weight 31, and twenty-two times greater for weight 40. These results show that the equal weights are more robust than the original weight sequence.

6.2.2 Traditional Approach. The linear program formulation was solved for the original weight sequences for both time blocks. The SAS Linear Programming Procedure produced the ranges for each objective function coefficient where the optimal solution remained unchanged. These ranges were used to produce the optimal ranges for weights of interest for each time block. The optimal ranges are shown in Tables 9 and 10 below.

Table 9 Classical Sensitivity Analysis Range for Time Block One Weights

Weight #	Original Value	Low Value	High Value
20	.203	4%	14%
22	.145	20%	6%
27	.203	3%	2%
30	.145	33%	8%
31	.203	13%	10%

Table 10 Classical Sensitivity Analysis Range for Time Block Six Weights

Weight #	Original Value	Low Value	High Value
9	.1491	10%	15%
20	.1491	23%	4%
27	.1491	3%	27%
30	.1355	28%	4%
31	.1897	11%	14%
40	.1355	2%	13%

The ranges shown in both tables above are larger, significantly larger in some cases, than the ranges produced by the tolerance approach. This is not surprising given the fact that the tolerance approach looks at all the basic variables at one time, whereas the classical approach looks at them individually.

6.2.3 Manual Approach. For this approach, the network problem was solved for various values of each weight above and below the original value to determine where the optimal

solution changed. The difference in this approach and the classical approach is that in this approach the weights were renormalized for each solution. In the classical approach only the change in an individual weight was determined. The results for time block one and six are shown in Tables 11 and 12.

Table 11 Manual Sensitivity Analysis Range for Time Block One Weights

Weight #	Original Value	Low Value	High Value
20	.203	1%	10%
22	.145	16%	4%
27	.203	6%	0%
30	.145	28%	5%
31	.203	11%	8%

Table 12 Manual Sensitivity Analysis Range for Time Block Six Weights

Weight #	Original Value	Low Value	High Value
9	.1491	10%	15%
20	.1491	23%	4%
27	.1491	3%	27%
30	.1355	28%	4%
31	.1897	11%	14%
40	.1355	2%	13%

These tables also show larger ranges for the weights of interest for both time blocks than the tolerance approach.

But, they are not as large as the ranges produced by classical sensitivity analysis.

6.3 EVAL Comparisons

This section compares the overall EVAL results for time blocks one and six to an average EVAL value calculated by the DOD. This average value is calculated from the EVAL value for a large number of random frequency assignments generated by the DOD. The average EVAL value is used to determine if the network formulated for this problem produces a better frequency assignment on average than just picking any random frequency assignment.

For time block one, the EVAL numbers were calculated from 10,000 random frequency assignments. The average for these assignments is 12.6148 and the standard deviation is 0.3265 (10). The average EVAL result generated for this project for time block one is 26.65. This is forty-three standard deviations above the mean of the random assignments which means the network frequency assignments are far superior to a random frequency assignment.

For time block six, 5050 random frequency assignments were generated to produce a mean of 10.1835 and a standard deviation of 0.3233. The average EVAL result for a time block six network solution is 23.87. This is more than forty-two standard deviations above the mean of the random assignments

and means the network is producing far more productive frequency assignments than could be expected randomly.

6.4 Conclusions

This section draws some conclusions from the results presented in the previous sections. The conclusions cover what affect increasing the weights of interest have on the overall and area geolocation values and which frequency assignment produced the most robust frequency assignments.

The same basic pattern emerged for all five of the weights investigated for time block one with regard to the EVAL overall and area geolocation values. Slightly increasing the weight from zero produces a small increase for the overall EVAL result but when the weight gets larger than approximately 0.3 the EVAL value starts to decrease at a fairly steep rate. The range for the area geolocation value is bigger than the overall value, approximately 0.0 to 0.5, but it to starts to decrease if the weight is larger than 0.5.

For time block six, no one statement can describe the behavior of increasing the weight values. For weight 9 and weight 20, increasing the weight value decreases the overall EVAL value and causes the area geolocation value to increase. Increasing the weight value for weight 27 causes the overall EVAL value to oscillate where large weight values strictly decrease the value. The same is true of the area geolocation

value for weight 27, it increases and decreases for small weight values and decreases for large values. Weights 30 and 31 have interesting effects on the two EVAL values. The overall value is flat for small values, increases for medium values, and is flat again for large values. The area geolocation values increase slightly for small weight values, decreases sharply for medium values, and is flat for large values. Lastly, weight 40 increases both the overall and the area geolocation values for EVAL as the weight increases.

Looking at the results from the sensitivity analysis generated by the tolerance approach, the weight sequence with equal weights for all areas produced the most robust frequency assignments for the network. This is true for both time block one and for time block six. In time block one, the original weight sequence produced no optimality range for the five weights where the equal weight sequence produced ranges for all five weights. For time block six, both sequences generated optimality ranges for all six weights, but the equal weight sequence produced ranges that were at most twenty-two times greater and at least two times greater than the original sequence ranges. Therefore, the equal weight sequence produced the most robust frequency assignments for both time blocks.

Additionally, based on the results from time blocks one and six, the other ten time blocks were evaluated for equal

weights. The purpose is to determine if the equal weight sequence is more robust for frequency assignments than the original weight sequences furnished by the DOD. The results showed that the equal weight sequences are the most robust for all time blocks. The optimality ranges for the weights of interest were much larger for most weights and at least as large for all weights of the equal weight sequence.

Finally, comparing the EVAL solution to the network solution as the weight values increase shows an interesting result. The network solution does not predict how the EVAL solution changes with respect to the weight changes. But, looking at the overall EVAL solutions and comparing them to random solutions generated by the DOD, the network generated for this project appears to produce good frequency assignments for the true nonlinear network.

6.5 Recommendations

Some areas of interest that merit investigation in future research projects are:

1. Investigate other weights for time blocks one and six to determine if the equal weight sequence is more robust than all weight sequences.

2. Investigate other time blocks for the weight sequences used in this project to determine how the time block data changes the weights effects on the EVAL results.

Appendix A

This appendix contains the solutions from the linear formulation of the sample problem in chapter four.

SOLUTION 1 - A PRIORI SOLUTION:

Optimal Solution = 0.058153

Receiving Station	Frequency Assignment
1	1
1	2
1	3
2	1
2	2
2	3
3	0
3	0
3	0
4	0
4	0
4	0
5	1
5	2
5	3

SOLUTION 2:

Optimal Solution = 0.112772   *A Priori* Solution = 0.112772

Station	Frequency	Station	Frequency
1	1	1	1
1	2	1	2
1	3	1	3
2	1	2	1
2	2	2	2
2	3	2	3
3	0	3	0
3	0	3	0
3	0	3	0
4	0	4	0
4	0	4	0
4	0	4	0
5	1	5	1
5	2	5	2
5	3	5	3

Percent Difference = 0%

SOLUTION 3:

Optimal Solution = 0.039313   *A Priori* Solution = 0.039313

Station	Frequency
1	1
1	2
1	3
2	1
2	2
2	3
3	0
3	0
3	0
4	0
4	0
4	0
5	1
5	2
5	3

Station	Frequency
1	1
1	2
1	3
2	1
2	2
2	3
3	0
3	0
3	0
4	0
4	0
4	0
5	1
5	2
5	3

Percent Difference = 0%

SOLUTION 4:

Optimal Solution = 0.042193   *A Priori* Solution = 0.041913

Station	Frequency		Station	Frequency
1	1		1	1
1	0		1	2
1	3		1	3
2	1		2	1
2	2		2	2
2	3		2	3
3	0		3	0
3	2		3	0
3	0		3	0
4	0		4	0
4	0		4	0
4	0		4	0
5	1		5	1
5	2		5	2
5	3		5	3

Percent Difference = 1%

SOLUTION 5:

Optimal Solution = 0.038847 A Priori Solution = 0.038847

Station	Frequency	Station	Frequency
1	1	1	1
1	2	1	2
1	3	1	3
2	1	2	1
2	2	2	2
2	3	2	3
3	0	3	0
3	0	3	0
3	0	3	0
4	0	4	0
4	0	4	0
4	0	4	0
5	1	5	1
5	2	5	2
5	3	5	3

Percent Difference = 0%

SOLUTION 6:

Optimal Solution = 0.156461   *A Priori* Solution = 0.156461

Station	Frequency	Station	Frequency
1	1	1	1
1	2	1	2
1	3	1	3
2	1	2	1
2	2	2	2
2	3	2	3
3	0	3	0
3	0	3	0
3	0	3	0
4	0	4	0
4	0	4	0
4	0	4	0
5	1	5	1
5	2	5	2
5	3	5	3

Percent Difference = 0%

SOLUTION 7:

Optimal Solution = 0.024101   *A Priori* Solution = 0.024101

Station	Frequency	Station	Frequency
1	1	1	1
1	2	1	2
1	3	1	3
2	1	2	1
2	2	2	2
2	3	2	3
3	0	3	0
3	0	3	0
3	0	3	0
4	0	4	0
4	0	4	0
4	0	4	0
5	1	5	1
5	2	5	2
5	3	5	3

Percent Difference = 0%

SOLUTION 8:

Optimal Solution = 0.030811   *A Priori* Solution = 0.028577

Station	Frequency	Station	Frequency
1	1	1	1
1	2	1	2
1	3	1	3
2	1	2	1
2	0	2	2
2	0	2	3
3	0	3	0
3	2	3	0
3	3	3	0
4	0	4	0
4	0	4	0
4	0	4	0
5	1	5	1
5	2	5	2
5	3	5	3

Percent Difference = 7%

SOLUTION 9:

Optimal Solution = 0.023261   A Priori Solution = 0.023261

Station	Frequency	Station	Frequency
1	1	1	1
1	2	1	2
1	3	1	3
2	1	2	1
2	2	2	2
2	3	2	3
3	0	3	0
3	0	3	0
3	0	3	0
4	0	4	0
4	0	4	0
4	0	4	0
5	1	5	1
5	2	5	2
5	3	5	3

Percent Difference = 0%

APPENDIX B

THIS PROGRAM WAS ADAPTED FROM KRISTA JOHNSON'S THESIS TO COMPUTE THE OBJECTIVE FUNCTION COEFFIENTS FOR THE FIRST OBJECTIVE FUNCTION AND IT CREATES THE INPUT FILE FOR SAS PROC NETFLOW.

```
      PROGRAM NET

      PARAMETER (I = 40,J = 30,K = 31)
      REAL F(I,K),RP(I,J,K),DP(I,J,K),COEF(J,K),W(I,J),U(I)
     &   LAMB1,LAMB2,SCALE,D,DIV
      INTEGER I1,J1,K1,CAP,COV,ECAP,MTOTAL,M(J),
     &        SUP(J,K),DEM(J,K),DEMAND

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                   C
C OPEN DATA FILES AND READ THE DATA                 C
C                                                   C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      OPEN (10,FILE='D1.DAT',STATUS='OLD')
      OPEN (11,FILE='F1.DAT',STATUS='OLD')
      OPEN (12,FILE='FAN.DAT',STATUS='OLD')
      OPEN (13,FILE='MM1.DAT',STATUS='OLD')
      OPEN (14,FILE='U.DAT',STATUS='OLD')
      OPEN (20,FILE='TEMP.SAS',STATUS='NEW')
      OPEN (21,FILE='TEMP.DAT',STATUS='NEW')

      CAP = 1
      COV = 2
      ECAP = 5
      LAMB1 =  .91
      LAMB2 = -.09
      SCALE =  .01
      D = LAMB2*SCALE

      DO 100 I1=1,I
         READ (11,500) (F(I1,K1), K1=1,K)
         READ (14,503) U(I1)
 100  CONTINUE

      DO 110 I1=1,I
         DO 105 J1=1,J
            READ (10,502) (DP(I1,J1,K1), K1=1,K)
 105     CONTINUE
 110  CONTINUE

      DO 115 I1=1,I
         DO 112 J1=1,J
            READ (12,501) W(I1,J1)
```

```
 112      CONTINUE
 115   CONTINUE

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                      C
C COMPUTE COEFFICIENTS FOR THE FIRST OBJECTIVE FUNCTION C
C AND CREATE FILE CVECTOR FOR SENSITIVITY ANALYSIS      C
C                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


       DO 140 J1=1,J
          DO 130 K1=1,K
             COEF(J1,K1) = 0.0
             DO 120 I1=1,I
                 COEF(J1,K1) = COEF(J1,K1)+F(I1,K1)
      &                     *DP(I1,J1,K1)*W(I1,J1)*U(I1)
 120         CONTINUE
          COEF(J1,K1) = LAMB1*COEF(J1,K1)
          WRITE (21,520) COEF(J1,K1)
 130      CONTINUE
 140   CONTINUE

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                        C
C COMPUTE SUPPLY AND DEMAND REQUIREMENTS FOR THE NETWORK  C
C                                                        C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


       MTOTAL = 0
       DO 160 J1=1,J
          READ (13,506) M(J1)
          MTOTAL = MTOTAL + M(J1)
 160   CONTINUE

       DEMAND = MTOTAL - K*COV


       DO 180 J1=1,J
          DO 170 K1=1,K
             IF (K1.EQ.1) THEN
                                  SUP(J1,K1) = M(J1)
                ELSE
                     SUP(J1,K1) = 0
             END IF

             IF (J1.EQ.30) THEN
                                 DEM(J1,K1) = COV
                ELSE
```

```
                 DEM(J1,K1) = 0
         END IF

 170     CONTINUE
 180  CONTINUE

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                     C
C CREATE SAS NETFLOW INPUT FILE                       C
C                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


      WRITE (20,*) 'TITLE ''WORLD-WIDE SENSOR SYSTEM'';'
      WRITE (20,*) 'OPTIONS LINESIZE=78;'
      WRITE (20,*) 'DATA XXX;'
      WRITE (20,*) 'INPUT INNODE $ OUTNODE $ MIN CAP COST SUP
     &                          DEM;'
      WRITE (20,*) 'CARDS;'
      WRITE (20,*)

      DO 200 J1=1,J
         DO 190 K1=1,K
            WRITE (20,510) J1, 'R', K1, 'F', '.', '1',
     &            COEF(J1,K1), SUP(J1,K1), DEM(J1,K1)
 190     CONTINUE
      WRITE (20,515) J1, 'R   S .  .          0  0  0'
 200  CONTINUE

      DO 210 K1=1,K
         WRITE (20,511) K1, 'F   E .  .', D, '0  0'
 210  CONTINUE

      DO 220 K1=1,K
         WRITE (20,512) K1, 'F   N .', ECAP, '0  0  0'
 220  CONTINUE

      WRITE (20,513) 'E   T .  .      0  0  0'
      WRITE (20,513) 'N   T .  .      0  0  0'
      WRITE (20,514) 'S   T .  .      0  0', DEMAND
      WRITE (20,*) ';'
      WRITE (20,*)
      WRITE (20,*) 'PROC NETFLOW MAXIMUM DATA=XXX',
     &                       'ARCOUT=SOLUTION;'
      WRITE (20,*) '      HEADNODE OUTNODE;'
      WRITE (20,*) '      TAILNODE INNODE;'
      WRITE (20,*) '      MINFLOW MIN;'
      WRITE (20,*) '      CAPACITY CAP;'
      WRITE (20,*) '      COST COST;'
      WRITE (20,*) '      SUPPLY SUP;'
```

```
      WRITE (20,*) '      DEMAND DEM;'
      WRITE (20,*) 'PROC PRINT DATA=SOLUTION;'
      WRITE (20,*) 'SUM _FCOST_;'

500   FORMAT(1X,31(F3.2,1X))
501   FORMAT(24X,F6.4)
502   FORMAT(1X,31(F3.2,1X))

503   FORMAT(F8.1)
506   FORMAT(I2)
508   FORMAT(1X,A14,1X,I3,1X,A1)
509   FORMAT(1X,A1,1X,I2,A1,1X,A1,1X,I2,1X,A1,I2,1X,I2)
510   FORMAT(1X,2(I2,A1,1X),A1,2X,A1,1X,F11.6,1X,2(I2,1X))
511   FORMAT(1X,I2,A10,2X,F6.4,2X,A4)
512   FORMAT(1X,I2,A7,1X,I2,7X,A7)
513   FORMAT(3X,A24)
514   FORMAT(3X,A21,1X,I3)
515   FORMAT(1X,I2,A28)
520   FORMAT(1X,F9.6)
530   FORMAT(3X,A1,2X,I2,A15,1X,I3,3X,A1)
531   FORMAT(3X,A1,2X,I2,A23)

      END
```

APPENDIX C

THIS PROGRAM CREATES THE LP FORMULATION OF THE WORLD-WIDE SENSOR SYSTEM NETWORK FOR SAS PROC LP IN THE SPARSE INPUT FORMAT.

```
      PROGRAM SPARSE

      PARAMETER (I = 40,J = 30,K = 31)
      REAL COEF(J,K)
      INTEGER I1,J1,K1,COV,COUNT,MAX,MXCOV,MNCOV,M1(J)
      CHARACTER*6 XNAME(J,K)
      CHARACTER*3 YNAME(K)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                  C
C OPEN DATA FILES AND READ THE DATA                C
C                                                  C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      OPEN (10,FILE='XNAME.DAT',STATUS='OLD')
      OPEN (11,FILE='YNAME.DAT',STATUS='OLD')
      OPEN (12,FILE='CVECTOR.DAT',STATUS='OLD')
      OPEN (13,FILE='MM1.DAT',STATUS='OLD')
      OPEN (20,FILE='SPARSE6.SAS',STATUS='NEW')

      MXCOV = 7
      MNCOV = 2
      MAX = 240
      COUNT = 100

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                       C
C READ VARIABLE NAMES                                   C
C                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      DO 110 J1=1,J
         DO 100 K1=1,K
            READ (10,500) XNAME(J1,K1)
 100     CONTINUE
 110  CONTINUE

      DO 120 K1=1,K
         READ (11,501) YNAME(K1)
 120  CONTINUE

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                      C
C READ OBJECTIVE FUNCTION COEFICIENTS                   C
C                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      DO 140 J1=1,J
         DO 130 K1=1,K
            READ (12,502) COEF(J1,K1)
```

```
 130      CONTINUE

          READ (13,503) M1(J1)
 140   CONTINUE


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                        C
C CREATE SAS LP INPUT FILE                               C
C                                                        C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

       WRITE (20,*) 'TITLE ''WORLD-WIDE SENSOR SYSTEM'';'
       WRITE (20,*) 'OPTIONS LINESIZE=78;'
       WRITE (20,*) 'DATA;'
       WRITE (20,*)
       WRITE (20,*) 'INPUT _TYPE_ $ _COL_ $ _ROW_ $ _COEF_;'
       WRITE (20,*)
       WRITE (20,*) 'CARDS;'
       WRITE (20,*)
       WRITE (20,*) 'MAX        .         DETECT    .'
       WRITE (20,*) 'UPPERBD    .         AVAIL     .'

       DO 145 J1=1,J
        WRITE (20,504) 'LE       .         CON',COUNT,'     .'
          COUNT = COUNT + 1
 145   CONTINUE

       DO 150 K1=1,K
        WRITE (20,504) 'LE       .         CON',COUNT,'     .'
          COUNT = COUNT + 1
 150   CONTINUE

       DO 160 K1=1,K
        WRITE (20,504) 'GE       .         CON',COUNT,'     .'
          COUNT = COUNT + 1
 160   CONTINUE

       DO 180 J1=1,J
          DO 170 K1=1,K
             WRITE (20,505) '.         ',XNAME(J1,K1),
      &            'DETECT', '     ',COEF(J1,K1)
             WRITE (20,506) '.         ',XNAME(J1,K1),
      &                 'AVAIL','     1'


 170      CONTINUE
 180   CONTINUE

       DO 190 K1=1,K
```

```
          WRITE (20,507) '.          ',YNAME(K1),
   &       'DETECT',' -.0009'
190   CONTINUE

      COUNT = 100

      DO 210 J1=1,J
         DO 200 K1=1,K
            WRITE (20,509) '.          ',XNAME(J1,K1),
   &                'CON', COUNT,'      1'
200      CONTINUE
      COUNT = COUNT + 1
210   CONTINUE

      DO 230 K1=1,K
         DO 220 J1=1,J
            WRITE (20,509) '.          ',XNAME(J1,K1),
   &                 'CON', COUNT,'      1'
220      CONTINUE
         WRITE (20,510) '.          ',YNAME(K1),'          CON',
   &           COUNT,'     -1'
         COUNT = COUNT + 1
230   CONTINUE

      DO 250 K1=1,K
         DO 240 J1=1,J
            WRITE (20,509) '.          ',XNAME(J1,K1),
   &                 'CON',COUNT,'      1'
240      CONTINUE
         COUNT = COUNT + 1
250   CONTINUE

      COUNT = 100


      DO 255 J1=1,J
         WRITE (20,512) '.          _RHS_      CON',COUNT,
   &             '       ', M1(J1)
         COUNT = COUNT + 1
255   CONTINUE


      DO 260 K1=1,K
         WRITE (20,512) '.          _RHS_      CON',COUNT,
   &                '      ', MXCOV
         COUNT = COUNT + 1
260   CONTINUE


      DO 270 K1=1,K
```

```
      WRITE (20,512) '.          _RHS_     CON',COUNT,
     &         '     ', MNCOV
      COUNT = COUNT + 1
270  CONTINUE


     WRITE (20,*) ';'
     WRITE (20,*) 'PROC LP SPARSEDATA RANGEPRICE '
     WRITE (20,*) '            MAXIT1=40000 MAXIT2=40000;'

500  FORMAT(A6)
501  FORMAT(A3)
502  FORMAT(F15.7)
503  FORMAT(I2)
504  FORMAT(1X,A23,I3,A6)
505  FORMAT(1X,A10,A6,A10,A4,F15.7)
506  FORMAT(1X,A10,A6,A9,A6)
507  FORMAT(1X,A10,A3,A14,A7)
508  FORMAT(1X,A10,A6,A8,A7)
509  FORMAT(1X,A10,A6,A7,I3,A6)
510  FORMAT(1X,A10,A3,A11,I3,A6)
511  FORMAT(1X,A30,I3)
512  FORMAT(1X,A23,I3,A5,I2)

     END
```

APPENDIX D

THIS PROGRAM COMPUTES THE PROBABILITY OF GEOLOCATING A DISTRESS SIGNAL FOR EACH AREA OF TRANSMISSION GIVEN AN OPTIMAL ASSIGNMENT OF FREQUENCIES.

```
      PROGRAM GEOLOCATION

      PARAMETER (I = 40,J = 30,K = 31)
      REAL F(I,K),DP(I,J,K),G(I),W(I,J),U(I),SUM
      INTEGER I1,J1,K1,OPT(J,K)

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                   C
C OPEN DATA FILES AND READ THE DATA                 C
C                                                   C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC


      OPEN (10,FILE='D6.DAT',STATUS='OLD')
      OPEN (11,FILE='F6.DAT',STATUS='OLD')
      OPEN (12,FILE='FAN.DAT',STATUS='OLD')
      OPEN (13,FILE='OPT6.DAT',STATUS='OLD')
      OPEN (14,FILE='UU.DAT',STATUS='OLD')
      OPEN (20,FILE='GEO.DAT',STATUS='NEW')


      SUM = 0.0


      READ (14,503) WEIGHTS
      DO 100 I1=1,I
         READ (11,500) (F(I1,K1), K1=1,K)
         READ (14,503) U(I1)
         U(I1) = U(I1)/WEIGHTS
 100  CONTINUE


      DO 120 I1=1,I
         DO 110 J1=1,J
            READ (10,502) (DP(I1,J1,K1), K1=1,K)
 110     CONTINUE
 120  CONTINUE


      DO 140 I1=1,I
         DO 130 J1=1,J
            READ (12,501) W(I1,J1)
 130     CONTINUE
 140  CONTINUE


      DO 150 J1=1,J
         READ (13,504) (OPT(J1,K1), K1=1,K)
 150  CONTINUE
```

```
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
C                                                        C
C COMPUTE THE PROBABILITY OF GEOLOCATION FOR EACH        C
C INTEREST AREA AND WRITE THEM TO AN OUTPUT FILE.        C
C                                                        C
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      DO 180 I1=1,I
         G(I1) = 0.0
         DO 170 J1=1,J
            DO 160 K1=1,K
               IF (OPT(J1,K1).EQ.1) THEN
                 G(I1) = G(I1)+F(I1,K1)*DP(I1,J1,K1)*W(I1,J1)
     &                                *U(I1)
               END IF
 160        CONTINUE
 170     CONTINUE
         WRITE (20,505) 'The Probability for location', I1,
     &                              'is',G(I1)
         SUM = SUM + G(I1)
 180  CONTINUE
      WRITE (20,506) 'The Total is', SUM

 500  FORMAT(1X,31(F3.2,1X))
 501  FORMAT(24X,F6.4)
 502  FORMAT(1X,31(F3.2,1X))
 503  FORMAT(F8.3)
 504  FORMAT(31(I1,1X))
 505  FORMAT(1X,A28,1X,I2,1X,A2,F9.6)
 506  FORMAT(1X,A12,1X,F10.6)

      END
```

APPENDIX E

PROGRAM TO PERFORM NETWORK TOLERANCE ANALYSIS

```
      PROGRAM TOLERANCE

      PARAMETER (I=40,J=30,K=31)
      INTEGER I1,I2,J1,K1,CNT,OPT(J,K),SEQ(K),T1,T2
      REAL CK(J,K),TAU(J*K),DUMMY,RED(J,K),TAUSTAR,
     &     RANGE(2000,2),TEMP,F(I,K),RP(I,J,K),DP(I,J,K),
     &     COEF(J,K),W(I,J),U(I),COEF2(J,K),U1HIGH,U1LOW,
     &     U1RANGE,TEMPHIGH,TEMPLOW,SUM

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                               C
C OPEN INPUT AND OUTPUT FILES                   C
C                                               C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      OPEN (10,FILE='CVECTOR6.DAT',STATUS='OLD')
      OPEN (11,FILE='NETT6.OUT',STATUS='OLD')
      OPEN (12,FILE='RED6.DAT',STATUS='OLD')
      OPEN (13,FILE='NETSEQ.DAT',STATUS='OLD')
      OPEN (14,FILE='NETTOL6.DAT',STATUS='NEW')
      OPEN (20,FILE='D6.DAT',STATUS='OLD')
      OPEN (21,FILE='F6.DAT',STATUS='OLD')
      OPEN (22,FILE='FAN.DAT',STATUS='OLD')
      OPEN (23,FILE='U.DAT',STATUS='OLD')

      SUM = 0.0
      CNT = 0
      TAUSTAR = 1.0

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                               C
C READ IN DATA AND SUM COST COEFFICIENTS        C
C                                               C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      DO 50 K1=1,K
         READ (13,502) SEQ(K1)
  50  CONTINUE

      DO 70 K1=1,K
         DO 60 J1=1,J
            READ (12,500) RED(J1,SEQ(K1))
  60     CONTINUE
  70  CONTINUE


      DO 300 I1=1,I
         READ (21,520) (F(I1,K1), K1=1,K)
         READ (23,523) U(I1)
```

```
 300   CONTINUE

       DO 310 I1=1,I
          DO 305 J1=1,J
             READ (20,522) (DP(I1,J1,K1), K1=1,K)
 305      CONTINUE
 310   CONTINUE

       DO 315 I1=1,I
          DO 312 J1=1,J
             READ (22,521) W(I1,J1)
 312      CONTINUE
 315   CONTINUE

       DO 340 J1=1,J
          DO 330 K1=1,K
             CK(J1,K1) = 0.0
             DO 320 I1=1,I
                CK(J1,K1) = CK(J1,K1)+F(I1,K1)*DP(I1,J1,K1)
     &                   *W(I1,J1)*U(I1)
 320         CONTINUE
 330      CONTINUE
 340   CONTINUE

       DO 120 J1=1,J
          READ (11,501) (OPT(J1,K1), K1=1,K)
 120   CONTINUE

       DO 140 J1=1,J
          DO 130 K1=1,K
             IF(OPT(J1,K1).EQ.1) THEN
                                         SUM = SUM + CK(J1,K1)
             END IF
 130      CONTINUE
 140   CONTINUE

       WRITE (14,600) 'THE SUM IS ', SUM


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                 C
C CALCULATE TAU STAR VALUES AND FIND MINIMUM      C
C TAU STAR VALUE                                  C
C                                                 C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

       DO 160 J1=1,J
          DO 150 K1=1,K
             CNT = CNT + 1
             IF(OPT(J1,K1).EQ.0) THEN
```

```
                    TAU(CNT) = ABS(RED(J1,K1))/(SUM +
     &                            CK(J1,K1))
            ELSE
                 TAU(CNT) = 1
            END IF
            IF(TAUSTAR.GT.TAU(CNT)) THEN
                                     TAUSTAR = TAU(CNT)
                                     T1 = J1
                                     T2 = K1
            END IF
 150     CONTINUE
 160  CONTINUE

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                C
C OUPUT MINIMUM TAU STAR VALUE                   C
C                                                C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      WRITE (14,601) 'THE MINIMUM TAU STAR VALUE IS ',TAUSTAR,
     &                  T1,T2

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                C
C CALCULATE AND OUTPUT RANGE FOR BASIC           C
C COEFFICIENTS TO REMAIN OPTIMAL                 C
C                                                C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      CNT = 1
      DO 180 J1=1,J
         DO 170 K1=1,K
            TEMP = CK(J1,K1) * TAUSTAR
            IF(OPT(J1,K1).GT.1) THEN
                  RANGE(CNT,1) = CK(J1,K1) - TEMP
                  RANGE(CNT,2) = CK(J1,K1) + TEMP
            ELSE
                  RANGE(CNT,1) = CK(J1,K1)
                  RANGE(CNT,2) = CK(J1,K1)
            END IF
            CNT = CNT + 1
 170     CONTINUE
 180  CONTINUE

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                C
C DETERMINE WEIGHT RANGES WHERE SOLUTION         C
C REMAINS OPTIMAL                                C
```

```
C                                                 C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      DO 440 J1=1,J
         DO 430 K1=1,K
            COEF(J1,K1) = 0.0
            DO 420 I1=1,26
               COEF(J1,K1) = COEF(J1,K1)+F(I1,K1)*DP(I1,J1,K1)
     &                       *W(I1,J1)*U(I1)
 420        CONTINUE
 430     CONTINUE
 440  CONTINUE

      DO 443 J1=1,J
         DO 442 K1=1,K
            DO 441 I1=28,I
               COEF(J1,K1) = COEF(J1,K1)+F(I1,K1)*DP(I1,J1,K1)
     &                         *W(I1,J1)*U(I1)
 441        CONTINUE
 442     CONTINUE
 443  CONTINUE

      CNT = 1
      DO 360 J1=1,J
         DO 350 K1=1,K
            COEF2(J1,K1) = F(27,K1)*DP(27,J1,K1)*W(27,J1)
            CNT = CNT + 1
 350     CONTINUE
 360  CONTINUE

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                         C
C COMPUTE AND OUTPUT RANGE FOR WEIGHT 1                   C
C                                                         C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC

      U1PERCENT = 1.0
      CNT = 1
      DO 380 J1=1,J
         DO 370 K1=1,K
            IF((OPT(J1,K1).EQ.2) .AND. (COEF2(J1,K1).
     &                          NE.0)) THEN
                U1TEMP = ((RANGE(CNT,2)-COEF(J1,K1))/
     &                           COEF2(J1,K1))
                IF(U1PERCENT.GT.U1TEMP) THEN
                     U1PERCENT = U1TEMP
                     WRITE (14,620) J1,K1,U1PERCENT
                END IF
            END IF
```

```
         CNT = CNT + 1
370   CONTINUE
380 CONTINUE

    U1HIGH = U1PERCENT
    U1LOW  = U(9) - (U1PERCENT-U(9))
    WRITE (14,603) 'RANGE FOR WEIGHT 009 IS ',U1LOW, 'TO ',
   &                           U1HIGH


500 FORMAT(F15.10)
501 FORMAT(31(I1,1X))
502 FORMAT(I3)
520 FORMAT(1X,31(F3.2,1X))
521 FORMAT(24X,F6.4)
522 FORMAT(1X,31(F3.2,1X))
523 FORMAT(F6.4)
600 FORMAT(1X,A11,F10.7)
601 FORMAT(1X,A30,F15.13,2X,I2,2X,I2)
603 FORMAT(1X,A24,F19.13,1X,A3,F19.13)
605 FORMAT(1X,I4,2X,I4,2X,F9.7,2X,F9.7,2X,F9.7,2X,F9.7,
   &              2X,F9.7)
610 FORMAT(I3,I3,1X,F9.7,1X,F9.7,1X,F9.7,1X,F9.7)
620 FORMAT(I3,I3,1X,F15.12)

    END
```

Bibliography

1. Bazaraa, Mokhtar, John Jarvis and Hanif Sherali. Linear Programming and Network Flows. New York: John Wiley and Sons, 1990.

2. Berman, Oded and David Simchi-Levi. "Finding the Optimal a Priori Tour and Location of a Traveling Salesman with Nonhomogeneous Customers," Transportation Science, 22: 148-154 (May 1988).

3. Berman, Oded and David Simchi-Levi. "Minisum Location of a Traveling Salesman," Networks, 16: 239-254 (Fall 1986).

4. Bertsimas, Dimitris J. "Traveling Salesman Facility Location Problems," Transportation Science, 3: 184-191 (August 1989).

5. Bertsimas, Dimitris J. "The Probabilistic Minimum Spanning Tree Problem," Networks, 20: 245-275 (May 1990).

6. Bertsimas, Dimitris J., Patrick Jaillet and Amedeo R. Odoni. "A Priori Optimization," Operations Research, 38: 1019-1032 (November-December 1990).

7. Dearing, Christopher and Alfred B. Marsh. "High Frequency Direction Finding (HFDF) Resource Allocation Problem Statement With Transmitter Importance Weighting", Distributed to students and faculty in the Department of Operational Sciences, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, Spring 1991.

8. Hillier, F. S., and G. J. Lieberman. Introduction to Mathematical Programming. New York: McGraw-Hill Book Company, 1990.

9. Jaillet, Patrick. "A Priori Solution of a Traveling Salesman Problem in Which a Random Subset of the Customers are Visited," Operations Research, 36: 929-936 (November-December 1988).

10. Johnson, Krista E. Frequency Assignments for HFDF Receivers in a Search and Rescue Network. Master Thesis. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March, 1990.

11. Merrill, David, Yupo Chan, and Thomas Schuppe. "A Probabilistic Multiple-Traveling-Salesman Facility-Location Problem," Working Paper (Draft), July 1990.

12. Phillips, Don T. and Alberto Garcia-Diaz. Fundamentals of Network Analysis. Illinois: Waveland Press, Inc, 1990.

13. Ravi, N., and R. E. Wendell. "The Tolerance Approach to Sensitivity Analysis in Network Linear Programming," Networks, 18: 159-171 (March-April 1988).

14. Simchi-Levi, David, and Oded Berman. "A Heuristic Algorithm for the Traveling Salesman Location Problem on Networks," Operations Research, 36: 478-484 (May-June 1988).

15. Steppe, Jean M. Locating Direction Finders in a Generalized Search and Rescue Network. Master Thesis. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, March, 1991.

16. Wendell, Richard E. "The Tolerance Approach to Sensitivity Analysis in Linear Programming," Management Science, 31: 564-578 (May 1985).

17. Yu, Po-Lung. Multiple-Criteria Decision Making. New York: Plenum Press, 1985.

Vita

Captain Joseph C. Imsand, Jr. was born October 4, 1962 in Mobile, Alabama. He graduated from Ben C. Rain High School in Mobile in 1981 and attended Auburn University on an Air Force ROTC scholarship, graduating with a Bachelor of Science in Computer Engineering in August 1985. Upon graduation, he was commissioned in the USAF and served his first tour of duty as a computer research scientist at the Air Force Human Resources Laboratory, Brooks AFB, Texas. There he was responsible for procuring new equipment and maintenance for the four mainframe computer systems supporting the laboratory's research and development mission. He entered the School of Engineering, Air Force Institute of Technology, in August 1990.

Permanent Address: 4000 Shana Dr
Mobile, AL 36605

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
March 1992

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE
An A Priori Multiobjective Optimization Model of a Search and Rescue Network

5. FUNDING NUMBERS

6. AUTHOR(S)
Joseph C. Imsand Jr., Captain, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GOR/ENS/92M-17

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The purpose of this study was to determine the most robust frequency assignment for a search and rescue network. The focus was to assign weights to the transmitter areas of the network to determine which weight sequence produced the most robust frequency assignment. The DoD furnished weight sequences for twelve two-hour time blocks. These weight sequences were compared to a weight sequence with all weights having equal value. Network and linear programming were used to solve this problem and generate frequency assignments for all weight sequences. Classical and tolerance sensitivity analysis were used to analyze the frequency assignmnents generated by the different weight sequences. The weight sequence with all weights of equal value produced the most robust frequency assignments for all time blocks.

14. SUBJECT TERMS
Search and Rescue Network, Network Programming, Linear Programming, Sensitivity Analysis, Tolerance Analysis

15. NUMBER OF PAGES
117

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev 2-89)
Prescribed by ANSI Std Z39-18
298-102

SUPPLEMENTARY

INFORMATION

ERRATA

for

An a priori multiobjective optimization model of a search-and-rescue network

by

Joseph C Imsand Jr

Masters thesis
(AFIT/GOR/ENS/92M-17)

Subsequent checks on the EVAL computer program has unveiled discrepancies about some figures. A previous run on a time-block-6 optimization-tasking indicated a value of 24. Subsequent check by the research sponsor yielded 11.7. Thus instead of being 43 standard deviation from the random tasking result, the optimization-tasking is closer to 4-6 standard deviations, as indicated in previous research.

This inconsistency does not affect the overall finding of the study, which suggests that the optimization solutions do not track the EVAL solution changes with respect to the weighting schemes.